

# WebTed

## Ein System für Webbasierte Telediagnostik

Inauguraldissertation  
zur Erlangung des akademischen Grades  
eines Doktors der Naturwissenschaften  
der Universität Mannheim

vorgelegt von  
MARTIJN W. DE BOER  
aus Chêne Bourgeries  
Mai, 2000

Dekan: Professor Dr. Guido Moerkotte, Universität Mannheim  
Referent: Professor Dr. Reinhard Männer, Universität Mannheim  
Korreferent: Professor Dr. Wolfgang Effelsberg, Universität Mannheim

Tag der mündlichen Prüfung: 27. Juli 2000

*Für Silvia*





# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
<b>2</b>	<b>Grundlagen und Stand der Forschung</b>	<b>5</b>
2.1	Einführung . . . . .	6
2.2	Architektur . . . . .	8
2.3	Telediagnose . . . . .	9
2.4	Teletherapie . . . . .	18
2.5	Teleeducation . . . . .	21
2.6	Teleadministration . . . . .	22
2.7	Telemonitoring . . . . .	22
2.8	Geschichtliche Entwicklung . . . . .	23
2.9	Rahmenbedingungen . . . . .	25
2.10	Zusammenfassung . . . . .	28
<b>3</b>	<b>Architektur</b>	<b>29</b>
3.1	Anforderungen . . . . .	29
3.2	Lösungsansatz . . . . .	34
3.3	Client/Server im Web . . . . .	35
3.4	Zwischenergebnis . . . . .	43
3.5	Implementierung . . . . .	44
3.6	Telepathologie: ein Überblick . . . . .	52
3.7	Zusammenfassung . . . . .	55

<b>4</b>	<b>Statische Telediagnostik (StaTed)</b>	<b>57</b>
4.1	Java Applet . . . . .	59
4.2	Datenbankdienste . . . . .	61
4.3	Bildaufnahme . . . . .	70
4.4	Bildkompression und Bilddarstellung . . . . .	71
4.5	Whiteboard . . . . .	74
4.6	Zusammenfassung . . . . .	78
<b>5</b>	<b>Dynamische Telediagnostik (DynTed)</b>	<b>79</b>
5.1	Strategie zur Bildaufnahme . . . . .	80
5.2	Anforderungen . . . . .	82
5.3	Implementierung . . . . .	83
5.4	Zusammenfassung . . . . .	92
<b>6</b>	<b>Quantitative Telediagnostik (QuanTed)</b>	<b>93</b>
6.1	Aufbau des Kapitels . . . . .	97
6.2	Übersicht QuanTed . . . . .	97
6.3	Stand der Forschung . . . . .	99
6.4	Ansatz . . . . .	106
6.5	Realisierung . . . . .	109
6.6	Anwendungen . . . . .	127
6.7	Klassifikation . . . . .	135
6.8	Zusammenfassung . . . . .	135
<b>7</b>	<b>Diskussion und Ausblick</b>	<b>139</b>
<b>8</b>	<b>Zusammenfassung</b>	<b>149</b>
<b>A</b>	<b>Übersicht über den Server</b>	<b>151</b>
<b>B</b>	<b>Implementierte Glyphen</b>	<b>155</b>
B.1	Datenarchiv . . . . .	156
B.2	Rauschfilterung . . . . .	156
B.3	Segmentierung . . . . .	157

B.4 Datenanalyse . . . . .	158
<b>C Sicherheitskonzept</b>	<b>165</b>
<b>D IDL-Mikroskop Interface</b>	<b>169</b>
<b>E Glossar</b>	<b>171</b>
<b>Literatur</b>	<b>175</b>



# Kapitel 1

## Einleitung

Seit den fünfziger Jahren bemühen sich Wissenschaftler darum, Medizinern durch den Einsatz von Computern neue Möglichkeiten für Diagnose und Therapie zu geben. Beispiele sind etwa bildgebende Verfahren wie CT, NMR, Ultraschall oder Operationsplanungssysteme. Während erste Überlegungen zu den Einsatzmöglichkeiten von Computern in diesem Bereich schon sehr früh existierten, war die konkrete Umsetzung der Ideen von den verfügbaren Hard- und Softwareressourcen geprägt.

Durch die Verfügbarkeit von schnellen PCs und durch die Entwicklung des World Wide Web hat sich dieses Bild gewandelt: Das vergleichsweise einfache, offene Konzept des Internets, das vor rund 30 Jahren entworfen wurde, öffnet zusammen mit dem World Wide Web den Weg zum *Netcentric Computing*: Software wird nicht mehr lokal auf einen Computer installiert, sondern Anwendungen werden über das Web Benutzern zur Verfügung gestellt. Dadurch entfällt die aufwendige Installation von Software beim Anwender.

Der Wissensaustausch im medizinischen Umfeld wird dadurch in bisher ungekannter Weise gefördert: Der Zugriff auf vorhandenes Wissen in digitalen Bibliotheken ist unabhängig von Ort und Zeit geworden, Publikation von Texten und hochwertigen Bildern ist schneller und günstiger als in klassischen Printmedien möglich, die Kommunikation unter Kollegen erfolgt interaktiver und umfassender als bisher, und insbesondere können Dienste für die Verarbeitung von Daten aller Art über das World Wide Web angeboten werden.

Dies hat in allen Bereichen des Gesundheitswesens zu einer Aufbruchstimmung geführt, infolgederen Organisationen und Institutionen, aber auch Einzelpersonen ihre Dienste im Web bereitstellen. Das Angebot reicht von Gesundheitsportalen mit allgemeinen Informationsdiensten wie Medscape oder Multimedica, über Online-Bibliotheken bis hin zu Telemedizin-Konsultationsdiensten [Multimedica] [Medscape].

Gegenstand dieser Arbeit ist ein webbasierter Konsultationsdienst speziell für die Pathologie. Gegenwärtig basiert die Diagnostik in der Histopathologie, d.h. der Untersuchung von Geweben, hauptsächlich auf konventionellen Methoden der Lichtmikroskopie. Dazu werden einem Patienten operativ Gewebeproben entnommen und nach spezieller Präparierung durch einen Pathologen auf Merkmale wie eine veränderte Zellorganisation oder durch spezielle Marker erzeugte Färbungen untersucht [Riede 99]. Die Einschätzung einer Probe anhand dieser Merkmale stellt eine schwierige Aufgabe dar und wird fast unlösbar, wenn neben der Aussage über einen gut- oder bösartigen Tumortyp auch noch eine Prognose über den Verlauf der Krankheit gefordert wird. Dies liegt einerseits daran, daß Kriterien, die mit einer Prognose korrelieren, nicht unmittelbar aus der Fülle der Informationen eines mikroskopischen Bildes zu erkennen sind. Andererseits existieren für viele Tumorarten keine eindeutigen Kriterien, die einer solchen Prognose zugrunde gelegt werden könnten.

Am Deutschen Krebsforschungszentrum (DKFZ) in Heidelberg werden deshalb schon seit längerem Verfahren zur computergestützten Analyse von Tumormaterial entwickelt, die dem Pathologen ergänzende Informationen über das Gewebe geben. Computergestützte Bildauswertungsverfahren sind dem visuellen System des Menschen hinsichtlich Beschreibung, Objektivität und Reproduzierbarkeit überlegen und tragen so zu einer objektiveren Diagnose bei. In den letzten Jahren entstand deshalb eine Reihe von Verfahren zur Differenzierung von Tumorgewebe mit Hilfe der digitalen Bildverarbeitung. Einen Querschnitt bieten folgende Themen:

- Beschreibung der Zellkernmorphologie (zwei- und dreidimensional).
- Untersuchungen der Verteilung von Substrukturen im Zellkern.

- Beschreibung des Verhaltens von lebenden Zellen.
- Charakterisierung der Architektur von Tumorgewebe.
- Parametrisierung des strukturellen Aufbaus des Zytoskeletts einer Zelle.

Obwohl quantitative Parameter zur Beschreibung des Tumorgewebe, wie sie sich mit Methoden der digitalen Bildverarbeitung ermitteln lassen, für den Pathologen eine sinnvolle Ergänzung darstellen, haben sie bisher nur eingeschränkt Eingang in die histologische Praxis gefunden. Auch verfügbare Telepathologiesysteme bieten eine solche Funktionalität nicht. Deshalb wurde im Rahmen dieser Dissertation das WebTed (Webbasierte Telediagnostik)-System für Telepathologie aufgebaut, welches bisher verfügbare Systeme für die Telediagnostik um den Aspekt der *quantitativen Telepathologie*, d.h. der Möglichkeit, diagnostische Parameter aus histologischen Bildern zu extrahieren, erweitert.

Damit ergeben sich drei unterschiedliche Aspekte für die Telepathologie: Für die Telediagnostik werden entweder Bilder und klinische Daten einem Konsultanten bereitgestellt (statische Telepathologie), oder der Konsultant erhält Zugriff auf ein ferngesteuertes Mikroskop (dynamische Telepathologie). Der dritte Aspekt ist die quantitative Telepathologie, die über das WWW neue Dienste ermöglicht.

Gleichzeitig ändert die quantitative Telepathologie den Ansatz der Telemedizin: An die Stelle einer *Person-to-Person* Konsultation zwischen Konsultant und Referent, wie sie bei der üblichen Telekonsultation stattfindet, tritt ein sogenannter *Electronic Service (E-Service)*. Dem Benutzer werden Dienste angeboten, die er ohne Unterstützung durch einen Konsultanten nutzen kann. In diesem Fall sind es quantitative Parameter, die aus histologischen Schnittbildern extrahiert werden und so zusätzliche Hinweise für eine Diagnose geben. Dazu macht ein Pathologe digitale Aufnahmen von Gewebeschnitten und schickt diese über das Web an einen Server, der den E-Service bereitstellt. Auf dem Server werden die Daten mit Hilfe der digitalen Bildverarbeitung verarbeitet, und es werden quantitative Parameter zur

objektiven Beschreibung der Zellstrukturen errechnet. Diese Berechnungen erfolgen vollautomatisch und erfordern keinerlei menschliche Unterstützung.

Das in dieser Arbeit beschriebene WebTed-System unterstützt als erstes System alle drei Aspekte der Telepathologie: Die statische, die dynamische und die quantitative Telepathologie. Im Vergleich zu bisher verfügbaren Systemen ist es komplett webbasiert und benötigt außer einem aktuellen Webbrowser keine weitere Software auf dem Client.

Der Inhalt dieser Dissertation gliedert sich folgendermaßen: Kapitel 2 gibt einen Überblick über den Stand der Forschung auf dem Gebiet der Telemedizin und verfügbarer Telepathologiesysteme. Kapitel 3 evaluiert mögliche Client/Server-Architekturen, deren Eignung für die Verwendung im WWW, und es wird ein Framework für Telepathologie vorgestellt. Dieses Framework wird in Kapitel 4 für statische Telekonsultation, in Kapitel 5 für dynamische Telekonsultation und in Kapitel 6 für quantitative Telepathologie erweitert. Eine Diskussion und einen Ausblick beinhaltet das abschließende Kapitel 7.

Zum besseren Verständnis befindet sich in Anhang E (S. 171) ein Glossar mit den wichtigsten Abkürzungen.



# Kapitel 2

## Grundlagen und Stand der Forschung

Telemedizin befaßt sich mit der telematischen Erbringung medizinischer Dienstleistungen zwischen Ärzten oder zwischen Arzt und Patient, bei denen der ratsuchende Arzt/Patient und konsultierende Arzt räumlich voneinander getrennt sind. Das Spektrum der Anwendungen ist vielfältig, es reicht von einfachen Diensten wie z.B. der automatischen Kontrolle von Blutmeßwerten (Telemonitoring) über die Erstellung medizinischer Diagnosen (Telekonsultation) bis hin zur Telechirurgie (Teletherapie).

Zur Zeit beschäftigen sich weltweit mehr als 130 Projekte mit telemedizinischen Fragestellungen. Eine Übersicht über aktuelle Projekte findet sich im Onlineangebot der National Library of Medicine in den USA [Hayes 96] [TIE]. Während bei frühen Projekten die medizinischen Dienste nicht ohne Telemedizin zu erbringen waren und die Telemedizin damit eine absolute Notwendigkeit darstellte, stehen bei neueren Entwicklungen vor allem ökonomische Aspekte im Vordergrund. Ein Beispiel für frühe Anwendungen der Telemedizin waren z.B. die ersten bemannten Raumflügen der NASA, bei denen Astronauten in Audio- und Videokonferenzen bei der Bodenstation medizinische Hilfe anfordern konnten [NASA].

Im folgenden wird ein Überblick über verschiedene Anwendungen der Telemedizin mit dem Schwerpunkt Telepathologie gegeben.

## 2.1 Einführung

Die Medizin beschäftigt sich in erster Linie mit der Diagnose und Therapie von Krankheiten. Als Krankheit wird der Verlust des normalen Gesundheitszustandes mit seinen klinischen Manifestationen und spezifischen Ursachen verstanden. Der erste Schritt zur erfolgreichen Behandlung einer Krankheit ist die Erkennung der Ursachen, d.h. die Erstellung einer Diagnose. Grundlage für eine Diagnose bilden dabei Bilder unterschiedlicher Modalitäten (z.B. CT, NMR, Ultraschall, Histologie), Meßdaten (z.B. EKG), klinische Daten, aber auch Beobachtungsmerkmale, die durch den Tastsinn gewonnen werden (z.B. Erfühlen von Geschwülsten).

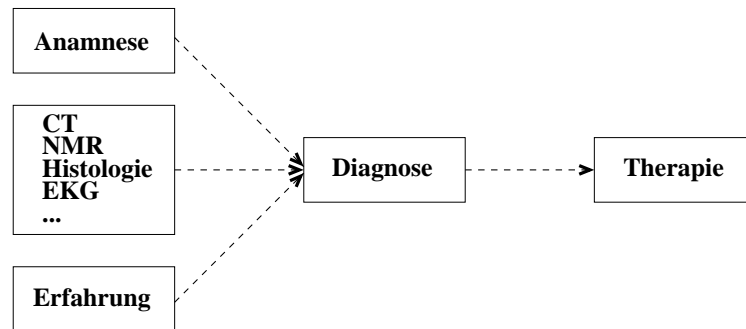


Abbildung 2.1: *Eine Diagnose beruht neben klinischen Daten und Meßdaten vor allem auf der Erfahrung des diagnostizierenden Arztes. Für die Durchführung einer Telediagnose ist es notwendig, die Daten einem entfernt sitzenden Konsultanten bereitzustellen.*

Um eine Diagnose telematisch durchzuführen, müssen dem Konsultanten die Bilddaten unterschiedlicher Modalitäten, Meßdaten und klinische Daten in geeigneter Form übermittelt werden. Es ist also notwendig, eine alternative Darstellung der Modalitäten zu finden, die sowohl telematisch übertragbar ist, als auch zu denselben Ergebnissen wie die ursprüngliche Modalität führt. Bei Bilddaten bedeutet dies eine Digitalisierung, die sich zu einem Konsultanten übermitteln und dort auf einem Monitor anzeigen läßt.

Im Idealfall sind Erfahrungen, die mit der ursprünglichen Modalität gewonnen wurden, auf die alternative Darstellung übertragbar. Die Erstellung einer Diagnose erfordert nämlich nicht nur ein fundiertes Wissen, sondern

auch ein hohes Maß an Erfahrung und Urteilstkraft. Ein effektiver Einsatz der Telemedizin ist deshalb nur möglich, wenn bisherige Erfahrungen weiterhin verwendbar sind. Dies ist von entscheidender Bedeutung für den erfolgreichen Einsatz der Telemedizin, denn letztlich ist das Ziel der Telemedizin eine telematische Anwendung der Erfahrung und des Wissens, um die gleiche Diagnosequalität erzielen zu können, als wenn der Spezialist direkt vor Ort wäre.

Während die telematische Durchführung von Diagnosen oder die telematische Überwachung von Meßwerten wie z.B. Blutdruck in den meisten Fällen mit überschaubarem Aufwand realisierbar ist, stellt sich der Fall der Teletherapie, etwa der Telechirurgie, sehr viel komplizierter dar. Es müssen Mittel bereitgestellt werden, um die Handgriffe eines Chirurgen telematisch zu übertragen und auszuführen. Dabei gilt bezüglich der Übertragbarkeit der Erfahrungen dasselbe wie schon bei der Telediagnose. Telechirurgie wird in Abschnitt 2.4.1 näher erläutert.

Heutzutage fällt unter den Begriff "Telemedizin" eine Vielzahl von Informations- und Konsultationssystemen. Sie lassen sich in fünf Kategorien einteilen:

**Telediagnose:** Die Erstellung von Diagnosen aufgrund von Bilddaten u.ä., welche in digitaler Form an einen Konsultanten geschickt werden, wird als Telediagnose bezeichnet. Eine spezielle Form hiervon ist das *Telekonsil*, d.h. die gemeinsame Konsultation von mehreren Fachkollegen zu einem Fall. Dabei hat jeder Teilnehmer während der Falldiskussion per Telefon- oder Videokonferenz auf seinem Computer Zugang zu allen Bildern und Daten eines Falles. Diese Art der Telemedizin wird z.B. in der Radiologie und Pathologie praktiziert.

**Teletherapie:** Die telematische Durchführung von Therapien, z.B. Operationen, ist Aufgabe der Teletherapie. Ein aktives Forschungsgebiet ist z.Zt. die Telechirurgie. In Zukunft wird sie eingesetzt, um Operationen direkt am Ort des Geschehens, z.B. in Krisengebieten, durchzuführen. Dazu werden mobile Einheiten mit einem fernsteuerbaren Roboter eingesetzt, die von einem im sicheren Gebiet sitzenden Operateur bedient

werden.

**Teleeducation:** Ziel der Teleeducation ist die einfache und schnelle Verbreitung von Informationen. Im Vergleich zu dem bisherigen Distributionsmedium, den wissenschaftlichen Zeitschriften, lassen sich über das WWW Beiträge kostengünstiger und schneller publizieren, da z.B. die hohen Kosten für den hochwertigen Druck farbiger Bilder entfallen.

**Teleadministration:** Die Verwaltung der Patientendaten, die Verteilung der Daten zwischen niedergelassenen Ärzten und Kliniken, die Abrechnung mit den Krankenkassen und die Fernwartung von medizinischen Geräten wird als Teleadministration bezeichnet. Sie kann zentral erfolgen und spart dadurch Personalressourcen.

**Telemonitoring:** Das Telemonitoring wird zur ortsunabhängigen Überwachung von Meßwerten eines Patienten im Heimbereich eingesetzt. Ein Beispiel hierfür ist das Projekt "Schlaganfall-Teleservice Saar". Per Funk übermittelt ein mit Sensoren versehenes Blutdruckmeßgerät über die Hausbasisstation Daten an den behandelnden Arzt. Sobald die Daten einen kritischen Zustand signalisieren, kontaktiert der Arzt den Patienten [FHG].

## 2.2 Architektur

Der allgemeine Aufbau eines Systems für Telemedizin ist in Abb. 2.2 dargestellt. Es besteht aus zwei Stationen, die über ein Netzwerk miteinander verbunden sind. Die Netzwerkverbindung ist dabei in der Regel eine ISDN- oder Internetverbindung, aber auch andere Verbindungen sind möglich (Satellitenverbindung). Von jeder Station können Daten (klinische Daten und Bilddaten) aus Datenquellen wie einem Archiv oder von bildgebenden Geräten (Mikroskop, CT, NMR, Sensoren) zusammengestellt und an eine andere Station verschickt werden.

Erweiterte Konzepte sehen dabei noch Komponenten für Telekonferenzen oder die interaktive Fernsteuerung von Geräten wie z.B. Mikroskopen (Te-

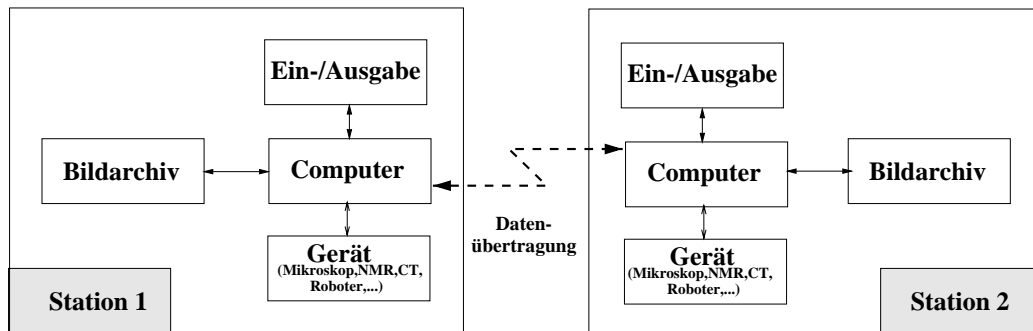


Abbildung 2.2: Abstrakter Aufbau eines Systems für Telemedizin mit zwei Stationen. An einer Station werden Daten zu einem Fall aus unterschiedlichen Quellen zusammengestellt und über ein Netzwerk an die andere Station zur Begutachtung verschickt.

lepathologie) oder Robotern (Telechirurgie) vor. Je nach Anwendungsgebiet sind an jede Station unterschiedliche Geräte angeschlossen; eine Sendestation für Telemonitoring hat z.B. nur ein Meßgerät, ein Modem und ist sehr viel kompakter als ein System für Telepathologie.

## 2.3 Telediagnose

Unter Telediagnose wird die telematische Erstellung von Diagnosen verstanden. Dazu werden einem *Konsultanten* (beratender Spezialist) die Daten zu einem Fall zur Begutachtung übermittelt. Grundlage für die Diagnose sind in der Regel klinische Daten und Bilddaten, die in digitaler Form übertragen werden. Bilddaten liegen entweder direkt in elektronischer Form vor (z.B. NMR) oder werden vorher, z.B. durch Scanner oder Kamera, digitalisiert. Eine besondere Form der Konsultation ist das *Telekonsil*, in dem ein Fall von mehreren Spezialisten (eventuell aus verschiedenen Fachbereichen) in einer gemeinsamen Telekonferenz diagnostiziert wird.

Es gibt zwei unterschiedliche Verfahren, um die Daten zwischen den Stationen zu verteilen. Bei dem einen werden alle zu einem Fall gehörigen Daten (klinische Daten und Bilddaten) in einem Paket an die Diagnosestation des Konsultanten geschickt (Abb. 2.3a), bei dem anderen Verfahren stellt ein Server die Bilder für den Konsultanten zum Abruf bereit und überträgt die

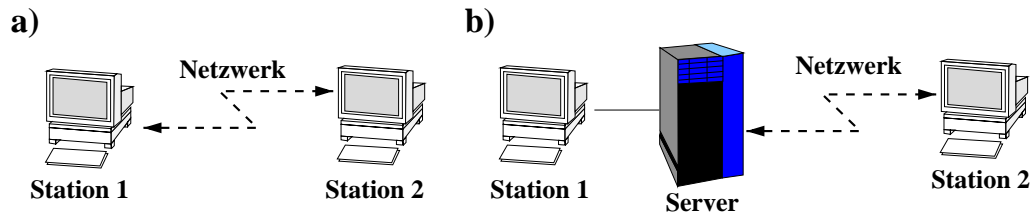


Abbildung 2.3: Möglichkeiten zur Verteilung der Daten für eine Telekonsultation.

Daten erst während der Konsultationssitzung (Abb. 2.3b). Beide Verfahren haben Vor- und Nachteile, an dieser Stelle sind Aspekte wie Bandbreite, Firewalls und Sicherheitsaspekte zu nennen.

### 2.3.1 Teleradiologie

Teleradiologie ist z.Zt. die am weitesten verbreitete Form der Telemedizin, sie wird heutzutage in vielen Kliniken routinemäßig eingesetzt. Es gibt eine Vielzahl von Gründen für den Einsatz von Teleradiologie. Zu den wichtigsten gehören die schnelle Versorgung von Patienten in Notfallsituationen, die Reduzierung der Anzahl von Krankentransporten, die Möglichkeit zur Expertenkonsultation sowie ökonomische Aspekte durch eine höhere Auslastung von Geräten und Personal.

Die Teleradiologie erlaubt den elektronischen Versand von Röntgen-, CT-, Ultraschall- und NMR-Daten für weitergehende Konsultationen in Spezialzentren. In den Spezialzentren begutachten Experten die Bilddaten auf einer Teleradiologie-Workstation und schicken die Diagnose an den Referenten (anfragenden Arzt) zurück. Je nach eingesetzter Software gibt es auch die Möglichkeit, den Fall über eine Telekonferenz online mit einem Kollegen zu besprechen oder 3D-Daten zu visualisieren.

Das Verfahren der Teleradiologie ist in Abb. 2.4 beschrieben. Aus unterschiedlichen Quellen (Archiv, CT-, NMR Geräte) trägt der Referent die Daten für die Konsultation zusammen. Diese werden zum Versand vorbereitet und an die Radiologiestation des Konsultanten geschickt. Der Konsultant erstellt eine Diagnose und schickt sie dem Referenten.

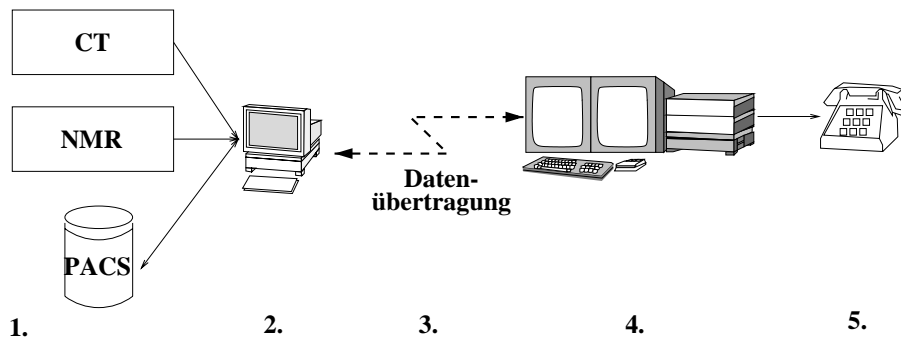


Abbildung 2.4: *Teleradiologie: Aus verschiedenen Datenquellen werden die Daten zu einem Fall zusammengestellt (1), zum Versand vorbereitet (2), zur Station des Konsultanten übertragen (3) und von einem Radiologen begutachtet (4). Nach der Befundung teilt der Konsultant dem Referenten seine Diagnose mit (5).*

Unmittelbare Kosteneinsparungen resultieren aus dem Wegfall der Transportkosten von Patienten, der Einsparung von Röntgenfilmen und insbesondere einer besseren Auslastung von Geräten und Personal. Eine Evaluation eines in Mecklenburg-Vorpommern eingesetzten Systems für die Diagnose von Akutpatienten ergab eine Kostenersparnis von 2,12 Mio. DM bei einer Investition von 0,35 Mio. DM in einem Zeitraum von fünf Jahren [Burchert 99]. In [Boland 96] wird beim routinemäßigen Einsatz von Teleradiologie von einem Kostenvorteil von US \$1,03 bei Röntgenbildern und US \$20,11 bei CT- oder NMR-Aufnahmen ausgegangen (unter Berücksichtigung der Materialkosten, Versand bzw. Übertragungskosten).

Von allen teleradiologischen Verfahren hat sich die Teleradiologie am schnellsten entwickelt. Hierfür sind verschiedene Gründe auszumachen: Zum einen liegen einige Bildmodalitäten (NMR, CT) bereits in digitaler Form vor, zum anderen wurde durch die Rationalisierung (Stichwort digitales Krankenhaus) ein Großteil der für die Teleradiologie notwendigen IT-Infrastruktur geschaffen. Dies betrifft die Entwicklung von Standards der Bilddigitalisierung und Kommunikation, bekannt unter *Picture Archiving and Communication Systems* (PACS), von Krankenhausinformationssystemen (*Hospital Information Systems*, HIS), und des DICOM<sup>1</sup> Standards für die Radiologie

<sup>1</sup>DICOM: Digital Imaging and Communication in Medicine.

[DICOM 3.0]. In diesem Standard sind die Austauschprotokolle zwischen unterschiedlichen Geräten (CT, NMR) und die Speicherformate festgelegt. Dadurch ist es mit geringem Aufwand möglich, Daten von einer DICOM Station an eine andere Station zu verschicken. In der Praxis scheitert dies leider an proprietären Erweiterungen des DICOM Standards, wie sie von einigen Herstellern vorgenommen wurden. Aufgrund der großen Anzahl von Teleradiologiesystemen wird hier auf eine Auflistung verzichtet.

### 2.3.2 Telepathologie

Die Pathologie beschäftigt sich mit der naturwissenschaftlichen Beschreibung der Ursachen, Entstehungsmechanismen und morphologischen Manifestationen von Krankheiten. Krankheiten verursachen Störungen auf der Organ-, Zell- oder Organellebene, die sich von Pathologen aufgrund der veränderten Zellstrukturen erkannt werden und in unterschiedliche Malignitätsgrade einstufen lassen.

Für die Befundung werden Gewebeproben, Tumorproben oder Blut auf dünnen Glaträgern aufgetragen und durch Fixierung und Färbung präpariert. Diese Proben werden dann unter dem Mikroskop bei unterschiedlichen Vergrößerungen (Objektiv  $16\times$ – $100\times$ ) betrachtet und auf Abweichungen der Zellstrukturen von der Norm hin beurteilt.

Die Grundlage für die Telepathologie bilden wie bei der Pathologie Gewebeschnitte, allerdings ist es unmöglich, eine hochauflösende Aufnahme eines ganzen Präparats analog zu Röntgenbildern bei der Teleradiologie zu verschicken, sie wäre selbst bei einer hohen Kompressionsrate noch 600 MB groß<sup>2</sup> und so nicht in kurzer Zeit übertragbar (zum Vergleich: ein hochauflösendes Röntgenbild hat eine Größe von ca. 25 MB). Zur Lösung des Problems werden zwei Strategien verfolgt: Entweder werden hochauflösende Bilder von Ausschnitten des Präparats übertragen (statische Telepathologie), oder der Konsultant verwendet ein ferngesteuertes Mikroskop bei dem

---

<sup>2</sup>Ein Präparat hat die Größe von  $1\text{cm}\times 2\text{cm}$ . Bei Verwendung eines  $25\times$  Objektivs werden  $183 \times 124 = 22.692$  Aufnahmen je  $768\times 576$  Pixel benötigt, um das ganze Präparat aufzunehmen. Daraus ergibt sich eine Datenmenge von etwa 30 GB. Nach einer Kompression der Bilddaten mit 1:50 ergibt sich eine Datenmenge von 600 MB.



ein komprimiertes Videobild der am Mikroskop angeschlossenen Videokamera übertragen wird [Kayser 99].<sup>3</sup>

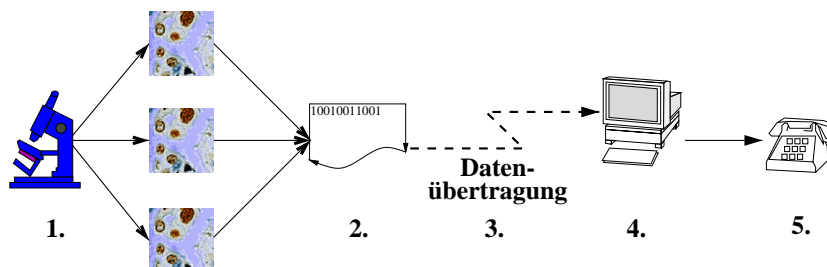


Abbildung 2.5: *Statische Telepathologie: Von einem Präparat werden repräsentative Ausschnitte ausgesucht (1), in einem Dokument zusammengestellt (2) und zur Befundung zum Konsultanten übertragen (3). Der Konsultant diagnostiziert den Fall (4) und teilt dem Referenten den Befund mit (5).*

Bei der statischen Telepathologie wählt der Referent fragliche Ausschnitte aus und macht mit einer am Mikroskop angeschlossenen Videokamera-Aufnahmen von den Ausschnitten. Diese werden dann entweder auf einem Server gespeichert und zum Abruf bereitgehalten (Abb. 2.3a), oder aber als digitales Dokument, das klinische Daten und histologische Bilder enthält, an den Konsultanten übertragen. Der Konsultant begutachtet den Fall und teilt dann die Diagnose mit. Falls das erhaltene Bildmaterial nicht ausreicht, müssen Bilder von weiteren Ausschnitten in einer zeitraubenden Prozedur angefragt werden.

Letzteres lässt sich mit der dynamischen Telepathologie umgehen; sie erlaubt dem Konsultanten die komplette Kontrolle über das Mikroskop (Telemikroskopie). Damit lassen sich die unterschiedlichen Funktionen des Mikroskops (Vergrößerung, Ausschnitt, Fokus, Licht etc.) vom Konsultanten fernsteuern. Die dynamische Telepathologie wird in Situationen eingesetzt, in denen kein Pathologe vor Ort ist, der wie bei der statischen Konsultation repräsentative Ausschnitte heraussuchen könnte. Dies ist z.B. bei folgendem Szenario der Fall: Nach der Entnahme einer Gewebeprobe während einer Tumoroperation an einer Klinik, an der kein Pathologe zur Verfügung steht,

<sup>3</sup>Dynamische Telepathologie wird auch als *store-and-forward telepathology* oder *off-line telepathology*, statische Telepathologie als *robotic telepathology* oder *on-line telepathology* bezeichnet.

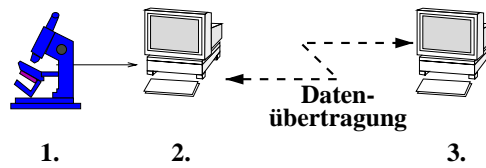


Abbildung 2.6: *Dynamische Telepathologie: Der Konsultant bedient mit seinem Rechner (3) ein ferngesteuertes Mikroskop (1). Das Videobild der am Mikroskop angeschlossenen Kamera wird in Echtzeit zum Rechner des Konsultanten übertragen. Ansteuerung des Mikroskops und Bildaufnahme erfolgt über einen Steuerrechner (2).*

wird die Gewebeprobe präpariert und unter das ferngesteuerte Mikroskop gelegt. Die Präparation erfordert keinen Pathologen und kann von einer MTA<sup>4</sup> durchgeführt werden. Mit der Telemikroskopie-Gegenstation übernimmt nun der Pathologe die Kontrolle über das Mikroskop um eine Schnelldiagnose zu erstellen.

Beide Verfahren haben Vor- und Nachteile. Die dynamische Telepathologie ermöglicht die freie Auswahl der zu betrachtenden Region, während diese bei der statischen Telepathologie fest vorgegeben sind. Dafür liefert die statische Telepathologie i.d.R. größere Bilder mit einer höheren Qualität, während dynamische Telepathologie sich oftmals auf den MPEG Standard beschränkt und eine wesentlich höhere Bandbreite (etwa 400 kBaud) benötigt [Kayser 99]. Die statische Telepathologie wird deshalb für Expertenkonsultationen eingesetzt, d.h. bei Konsultationen zwischen Spezialisten; die dynamische Pathologie findet Anwendung wenn kein Pathologe vor Ort ist, der die Auswahl der Ausschnitte übernehmen könnte.

Die Selektion der für eine Diagnose verwendeten Bildausschnitte ist für die Qualität der Diagnose von großer Bedeutung. Weinstein hält statische Telepathologie für ungeeignet, da die Selektion des Bildmaterials immer unvollständig und subjektiv ist und einen Konsultanten zu falschen Annahmen verleiten kann [Weinstein 90].

Nichtdestoweniger haben Untersuchungen zur Diagnosequalität ergeben, daß sich mit statischer Telepathologie bei Expertenkonsultationen eine mit

---

<sup>4</sup>MTA: Medizinisch Technische Assistentin.

der gewöhnlichen Pathologie vergleichbare Diagnosequalität erreichen läßt. Werden die Bildausschnitte aber nicht durch einen Pathologen ausgesucht, kann keine zuverlässige Diagnose gestellt werden. Im Falle von Experten-konsultationen wird eine Übereinstimmung von mehr als 90% erreicht, bei der dynamischen Telepathologie wird von einer ähnlich hohen Übereinstimmung berichtet [Callas 97] [Weinberg 96] [Strauss 99] [Della 98]. Meer hat eine Studie über den Einfluß der Bildkompressionen auf die Diagnosequalität durchgeführt und festgestellt, daß sich mit JPEG-komprimierten Bildern gute Ergebnisse erreicht werden [Meer 97].

### Standards

Aufgrund der geringen Verbreitung der Telepathologie haben sich noch keine Standards für die Kommunikation zwischen unterschiedlichen Systemen etabliert. Obwohl ursprünglich als Standard für Radiologie entwickelt, versteht sich die aktuelle Version von DICOM als Standard für die medizinische Bildgebung im allgemeinen und ist deshalb durch die Erweiterung durch Ergänzungen (sog. *Supplements*) auch für andere Disziplinen von großem Interesse. Speziell für die Pathologie ist dies Supplement 15 (Visible Light Image for Endoscopy, Microscopy and Photography), das z.Zt. als Vorschlag bei der ACR/Nema vorliegt [DICOM VL].

Von europäischer Seite wird im Rahmen des *EUROpean Pathology Assisted by Telematics for Health* (EUROPATH)-Projekts eine Erweiterung zum DICOM VL Standard für dynamische Telepathologie entworfen, in dem zusätzlich Schnittstellen zur Ansteuerung von motorisierten Mikroskopen, Dateiformaten und Kommunikationsprotokollen festgelegt werden [EUROPATH] [Klose 99]. Dieser Standard wird zum DICOM VL Standard kompatibel sein und den Anschluß an PACS<sup>5</sup> Systeme ermöglichen.

### Verbreitung

Im Gegensatz zur Teleradiologie, die bereits routinemäßig eingesetzt wird, verbreitet sich die Telepathologie nur schleppend [Wheeler 99]. Hierfür gibt

---

<sup>5</sup>PACS: Picture Archiving and Communication System

es mehrere Gründe: In der Pathologie werden Computer nur vereinzelt benutzt; die für Bildaquisition, Archivierung und Verarbeitung notwendige Infrastruktur muß deshalb noch entwickelt und ihre Handhabung den Pathologen vermittelt werden. Hingegen ist das grundsätzliche Interesse an der Telepathologie groß, wie z.B. eine Umfrage unter österreichischen Pathologen ergab [Mairinger 98]. Dazu kommt, daß von vielen Pathologen nur die dynamische Telepathologie akzeptiert wird. Die Verwendung von proprietären Kommunikationsprotokollen macht es zudem erforderlich, daß für eine Telekonsultation identische Software eingesetzt wird.

Von kommerzieller Seite bieten alle führenden Mikroskophersteller (Leica, Leitz, Olympus, Nikon) Systeme mit proprietären Standards für dynamische Telepathologie an. Für eine Konsultation ist es deshalb erforderlich, daß Konsultant und Referent identische Software einsetzen. Webbasierte Systeme werden nicht angeboten.

Von wissenschaftlicher Seite wird in verschiedenen Projekten an Telepathologie gearbeitet. Im EUROPATH-Projekt der EU arbeiten Mikroskophersteller und Universitäten daran, eine internetbasierte Kommunikationsplattform zum Austausch von Daten und Wissen zu erstellen [EUROPATH]. Nach Ablauf des Projekts existiert ein internetbasiertes System zum Austausch von (statischen) Multimediadokumenten (Pathconsult) und ein an der Universität Stuttgart entwickeltes System zur dynamischen Telepathologie [HISTKOM]. An der Charite in Berlin ist ein webbasiertes System für (eingeschränkt) dynamische Telepathologie im Einsatz. Zwar lassen sich alle wesentlichen Funktionen fernsteuern, aber zur Kompression und Übertragung werden pro aufgenommenem Bild etwa 4 Sekunden benötigt [Kayser 99]. In Italien wird ein Email-basiertes System für statische Telepathologie eingesetzt [Della 98]. Ein weiteres System für dynamische Telepathologie wurde an der University of Alabama in den USA entwickelt [TelePath 97].

Bis auf dem in Berliner Charite-Krankenhaus eingesetzte System sind alle Systeme für dynamische Telepathologie windowsbasiert und für den Einsatz im Internet ungeeignet.

### **Zukünftige Entwicklungen**

Eine wichtige Informationsquelle für die Diagnose und Behandlung von Tumoren ist die quantitative Bestimmung von Merkmalen bezüglich der Veränderung von Zellanordnungen, der Anfärbung von Kernen und der Markierung von Zellen mit Methoden der Molekularbiologie. Quantitative Parameter dienen der Objektivierung der Diagnose: Die Analyse histologischer Bilder ist mit einer Abstrahierung verbunden, bei der es zu einer subjektiven Selektion, und Zuordnung von Merkmalen kommt. Das visuelle System des Menschen ist durch die Vielfalt der möglichen Merkmale überfordert. Als Folge davon sind die Ergebnisse einer Untersuchung subjektiv und daher nicht unbedingt reproduzierbar.

Eine Lösung für die oben genannten Probleme stellt die computergestützte Morphometrie dar: Aus einem histologischen Bild werden per digitaler Bildverarbeitung Objekte detektiert, von denen charakteristische Parameter berechnet werden. Die Berechnung solcher Parameter ist reproduzierbar; allerdings ist das menschliche Auge dem Computer in Sachen Objekterkennung überlegen, bei der Zuordnung von Merkmalen mit kontinuierlichen Übergängen und der Einschätzung von Häufigkeitsverteilungen aber unterlegen.

Es werden jedoch nur an wenigen Forschungszentren Methoden entwickelt, um charakteristische Merkmale aus histologischen Bildern zu extrahieren. Die Präparierung von histologischen Schnitten zur Durchführung solcher Messungen ist ohne großen Aufwand erlernbar und öffnet damit die Möglichkeit für die Verbreitung mittels quantitativer Telepathologie: Eine Probe wird in einem Labor präpariert und mit einer digitalen Kamera fotografiert. Für eine quantitative Auswertung werden die Daten an einen Server für quantitative Telepathologie verschickt. Der Server führt mittels digitaler Bildverarbeitung eine Analyse durch und schickt die Ergebnisse an das Labor zurück.

Ein Beitrag zur Erweiterung der Telepathologie um eine Komponente für quantitative Telepathologie wird durch die vorliegende Arbeit geleistet. Das hier vorgestellte System erlaubt zum einen statische und dynamische Telepathologie, zum anderen ist es möglich, mit einer Komponente zur quantitati-

ven Telepathologie z.B. quantitativ die Struktur von Zellkernen zu analysieren (Kerngrading). Eine integrierte Entwicklungsumgebung für die digitale Bildverarbeitung erlaubt den schnellen Entwurf und die schnelle Evaluierung neuer Analyseverfahren. Das System zur quantitativen Telepathologie wird in Kapitel 6 vorgestellt.

### **2.3.3 Teledermatologie**

Teledermatologie wird für die Begutachtung von makroskopischen Bildern der Haut eingesetzt. Die Bilder werden mit einer handelsüblichen digitalen Kamera aufgenommen und per store-and-forward verschickt. Die Teledermatologie steht noch sehr am Anfang ihrer Entwicklung, im November 1999 fand die erste europäische Konferenz statt, auf der erste Erfahrungsberichte präsentiert wurden (vergl. [Loane 99]). In den USA setzt die US Army Systeme, bestehend aus Satellitentelefon, Laptop und digitaler Kamera, für die Teledermatologie ein [Crowther 95].

## **2.4 Teletherapie**

Ziel der Teletherapie ist die telematische Erbringung therapeutischer Leistungen. Beispiele sind Telepsychologie und Telechirurgie. Letztere ist Gegenstand der aktuellen Forschung, zu der im folgenden eine Übersicht gegeben wird.

### **2.4.1 Telechirurgie**

Die Telechirurgie setzt Virtual Reality Systeme zur Operationsplanung und Diagnostik ein, um durch Fernsteuerung eines Roboters Operationen telematisch durchzuführen. Zur Zeit sind noch keine Systeme im routinemäßigen Einsatz und werden auch sicherlich nur schwer vom Patienten akzeptiert werden. Systeme, die Teilaspekte eines Telechirurgiesystems implementieren, existieren bereits und werden in diesem Abschnitt vorgestellt. Die Motivation für den Einsatz solcher Systeme ist unterschiedlich. Während das

Interesse des Militärs daran liegt, direkt auf dem Schlachtfeld verwundeten Soldaten schnellstmöglich helfen zu können, um die Überlebenschance zu erhöhen (Stichwort golden hour), wird die Technologie im zivilen Bereich vor allem dazu verwendet, bessere medizinische Resultate zu erlangen:

**Präzision:** Der Chirurg verwendet ein Steuergerät, um den Roboter zu bedienen. Durch eine Übersetzung der Bewegungen (z.B. 5:1) wird neben der Anpassung an vergrößerte Ansichten eine größere Genauigkeit der Motorik erreicht. Zusätzlich lassen sich die zum Roboter übertragenen Bewegungen selektieren, um unbeabsichtigte Bewegungen (unruhige Hand) herauszufiltern.

Das Steuerpult kann optimal an den Chirurgen angepaßt werden, so daß ermüdende Haltungen vermieden werden und die Konzentration gesteigert wird.

**Schulung:** Bisher findet die Schulung von Chirurgen weitestgehend durch “training on the job” statt. Die Entwicklung von VR<sup>6</sup> Methoden erlaubt eine Schulung an Simulatoren, bei denen einem Operateur die Möglichkeit gegeben wird, einen Eingriff mit hoher Präzision zu trainieren, bevor er mit der Realität konfrontiert wird. Dadurch wird ein Beitrag zur Qualitätskontrolle und -sicherung in der Chirurgie geleistet. Nach einer erfolgreichen Schulung können dann nach “Fahrschulmodell” Operationen am reellen Objekt durchgeführt werden, bei denen im Notfall ein erfahrener Chirurg als “Fahrlehrer” die Kontrolle übernimmt.

**Minimal invasiv:** Die herkömmliche, minimal invasive Chirurgie leidet bei der Verwendung langer, starrer Instrumente durch den langen Abstand zwischen der Hand des Operateurs und dem operativ tätigen Werkzeug unter einer Verstärkung von unruhigen Bewegungen [Falk 99]. Der Einsatz von Robotiksystemen in Kombination mit geeigneten Informationssystemen beseitige solche Probleme.

---

<sup>6</sup>VR: Virtual Reality

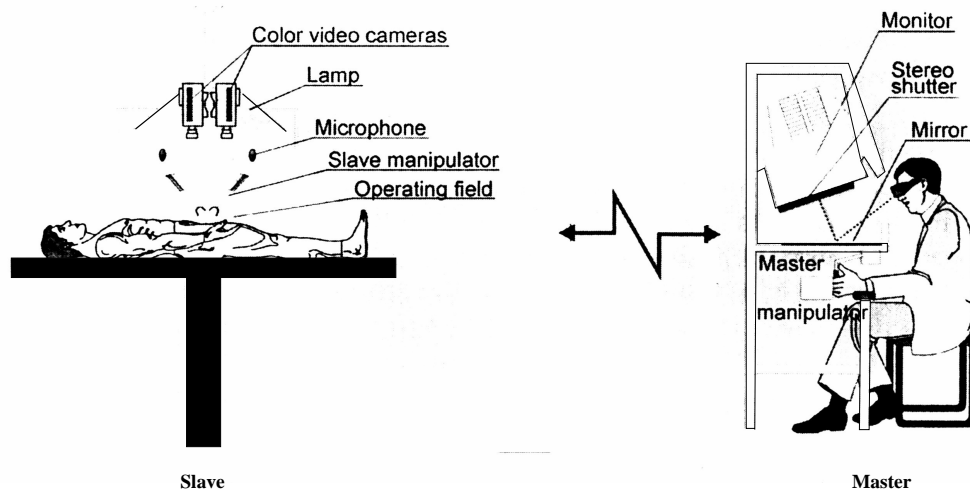


Abbildung 2.7: *Telechirurgie (aus [Kayser 99]).*

Der Aufbau eines solchen Systems ist in Abb. 2.7 gezeigt. Es ist aufgeteilt in *Master*- und *Slave*- Einheit. Die Master-Einheit dient als Steuerungskonsole für den Operateur, die Slave-Einheit befindet sich im OP-Raum. Die Slave-Einheit besteht aus zwei Farbkameras zur Stereoübertragung von Videobildern, Lampe und Roboterhand als Akteur für die Operation; die Master-Einheit besteht aus einem Steuergerät, mit dem der Manipulator bedient wird und einem Spiegel, auf den das Videobild projiziert wird. Um das Stereobild auf dem Spiegel zu sehen, trägt der Operateur eine Shutterbrille.

Die Master-Einheit kann entweder direkt, oder aber über eine Netzwerkverbindung (Telechirurgie) mit dem Slave verbunden sein. Ein Großteil des hier Vorgestellten ist heute noch Zukunftsmusik. Die wichtigsten Komponenten für die Telerobotik sind zum einen VR-Applikationen, die den Operateur auf geeignete Weise mit Informationen versorgen und nicht mit Bildern überfluten, zum anderen die Fernsteuerung geeigneter Manipulatoren für die Operation, sowie die computergestützte Navigation. Ein Teil dieser Komponenten befindet sich z.Zt. in Entwicklung, so daß klinikrelevante Ergebnisse nicht als autonome Systeme für die Telerobotik, sondern als Komponenten zur aktiven Unterstützung des Operateurs zu erwarten sind.



Zu den ersten Systemen, die Teilaspekte eines Telechirurgiesystems implementierten, gehört das von IBM und Integrated Surgical Systems entwickelte System ROBODOC zur Unterstützung des Operateurs bei Hüftoperationen. Mit diesem System kann ein Arzt die optimale Prothese auswählen und sie virtuell mit einem Computermodell einpassen. ROBODOC fräst dann nach Vorgabe des Arztes die notwendigen Stellen aus dem Hüftknochen heraus und erreicht den perfekten Sitz des Implantats. Im Vergleich zur konventionellen Methode wird die Paßgenauigkeit von 30% auf 98% erhöht [ROBODOC].

Ein Projekt, das den Aspekt eines gemeinsamen Konsiliums implementiert, ist das an der Robert-Rössle-Klinik durchgeführte Projekt OP 2000 [Schlag 99]. Es besitzt alle VR-Komponenten für eine Teleoperation, außer einem Akteur. Der Vorteil des Systems ist darin zu sehen, daß es eine Telepräsenz aller für das Konsilium notwendigen Ärzte erlaubt (Pathologen, Chirurgen, Radiologen etc.), es fehlt dabei aber an den geeigneten Manipulatoren, um eine Operation auch telematisch durchführen zu können.

## 2.5 Teleeducation

Das medizinische Wissen verdoppelt sich schätzungsweise alle fünf Jahre [Kayser 99]. Es ist daher unmöglich, dieses Wissen ständig an einem Ort präsent zu haben. Vielmehr ist es so, daß es eine Spezialisierung gibt, die an bestimmte Zentren und Personen gebunden ist. Ziel der Teleeducation ist es, dieses Wissen schnell und effektiv zu verbreiten. Das bisherige Distributionsmedium – die wissenschaftlichen Zeitschriften – wird dabei zunehmend durch webbasierte Informationsdienste verdrängt. Die Publikation im WWW hat einige Vorteile: Bilder können in hoher Qualität dargestellt werden, wie es sonst nur bei hochwertigen Printmedien möglich ist, und im Gegensatz zu Printmedien sind auch interaktive Elemente zum didaktischen Aufbau möglich. So ist es z.B. möglich, Datenbanken mit Musterfällen aufzubauen. Medizinstudenten können diese Fälle dann selbst diagnostizieren und ihre Ergebnisse mit den dort abgelegten Ergebnissen vergleichen.

Für Patienten und Mediziner gibt es im Internet eine Reihe von frei-

en und kommerziellen Informationsdatenbanken, die aktuelle Informationen für unterschiedliche Zielgruppen anbieten, siehe z.B. [CancerNet] [WebPath] [LINK] [E J Pathol].

## 2.6 Teleadministration

Die Verwaltung und Verteilung von Patientendaten zwischen niedergelassenen Ärzten, Kliniken und die Abrechnung mit den Krankenkassen sind die Aufgaben der Teleadministration. Ein Beispiel hierfür ist die Abrechnung der Daten zwischen Leistungserbringern und Kassen. Nach §301 Bundessozialgesetzbuch (BSGB) müssen seit dem 1.1.1997 alle Abrechnungsdaten in computerlesbarer Form bereitgestellt werden. Bisher erfolgte dies durch den Versand von Disketten, allerdings wird in einem Feldversuch der elektronische Versand dieser Daten erprobt. Die Daten werden dazu verschlüsselt und per ISDN an ein gemeinsam von allen Kassen betriebenes Rechenzentrum übermittelt.

Langfristiges Ziel der Teleadministration ist die zentrale Verwaltung aller Patientendaten in einer elektronischen Patientenakte, um unterschiedlichen Leistungserbringern eine umfassende Übersicht über den Patienten zu geben. Bei einem Notfall liegen dann alle Informationen über eventuelle Unverträglichkeiten von Medikamenten, Krankheiten etc. vor und erlauben so eine bessere Behandlung des Patienten. Das größte Problem bei der Realisierung der zentralen elektronischen Patientenakte stellen der Datenschutz und der Mißbrauch dieser Daten dar, es stellen sich Fragen wie z.B. *wem* sollen Daten gegeben werden, *welche* Daten sollen gegeben werden, *wie* wird verhindert, daß der Patient zur Herausgabe seiner Daten gezwungen wird etc.. Nähere Informationen finden sich z.B. in [Goetz 00a] [Goetz 00b] [Ulsenheimer 00].

## 2.7 Telemonitoring

Telemonitoring wird für die Überwachung von risikogefährdeten Patienten (z.B. chronisch Kranken, Diabetikern, Patienten mit Herzrhythmusstörungen)

eingesetzt. Mobile Untersuchungseinheiten registrieren Vitaldaten (Herzfrequenz, Temperatur, EKG) in der häuslichen Umgebung des Patienten. Die Daten werden zur Überwachung an eine Klinik geschickt, wo sie kontrolliert werden. Bei Komplikationen werden von der Klinik Hilfemaßnahmen eingeleitet.

Der Einsatz dieser Art von Fernüberwachung ermöglicht eine frühere Entlassung des Patienten aus der Klinik, und es wird ihm durch die ständige Überwachung ein großes Maß an Sicherheit gegeben. Der kürzere Aufenthalt im Krankenhaus und die Reduzierung von Arztbesuchen in Kombination mit Videovisiten bieten ein erhebliches Kosteneinsparungspotential. Ein Beispiel ist etwa das bereits erwähnte Beispiel “Schlaganfall Teleservice Saar” [FHG].

## 2.8 Geschichtliche Entwicklung

Die schnelle Verbreitung des Internets in den letzten 5 Jahren erweckt den Eindruck, daß es sich auch bei der Telemedizin um eine neuartige Entwicklung handelt. Historisch betrachtet hat im allgemeinen die Entwicklung einer Technologie und ihrer kommerziellen Verbreitung immer zu einem neuen Entwicklungsschub in der Telemedizin geführt. Die Ursprünge der Telemedizin reichen daher überraschend weit zurück, sie entstanden zeitgleich mit der Entwicklung des Telefons. Schon Anfang 1900 übermittelte Eindhoven EKGs mittels Telefon [Eindhoven 1906].

Einen bedeutenden Entwicklungsschub erfuhr die Telemedizin durch die bemannte Raumfahrt: Im All ist die nächste Möglichkeit zur medizinischen Versorgung tausende von Kilometern entfernt. Die NASA versah deshalb bereits die APOLLO-Missionen mit einer Ausstattung für eine für rudimentäre telemedizinische Versorgung. In den Raumanzügen waren verschiedene Sensoren zur Überwachung von physiologischen Parametern eingebaut, die diese Daten dann an die Bodenstation schickten. In akuten Notfällen gab es die Möglichkeit, Hilfe von der Bodenstation anzufragen und per Satellit Sprach- und Bilddaten zu übertragen. Da die Mission von den Computern der Bodenstation geleitet wurde, war der technische Mehraufwand für die Te-

lemedizin vergleichsweise gering, denn die Satellitenverbindungen zur Erde waren ohnehin Teil des Steuerungssystems.

In einem Buch von Rashid L. Bashur [Bashshur 75] von 1975 werden 15 aktive Telemedizinprojekte vorgestellt, die der Telemedizin neue Impulse gaben:

**STARPAHC (1972-1975):** Ziel des Space Technology Applied to Rural Papago Advanced Health Care (STARPAHC)-Projekts war die Verwendung der im All eingesetzten Telemedizinentechnologie zur medizinischen Versorgung der Einwohner im Papago Indian-Reservat (Arizona/USA). Das Indianerreservat wurde mit einer mobilen Versorgungseinheit mit medizinischen Geräten wie Elektrokardiographen und Röntgengeräten ausgestattet. Diese war über eine bidirektionale Mikrowellen-Funkverbindung mit einem Krankenhaus verbunden. Für die Kommunikation mit dem Spezialisten im Krankenhaus waren zwei hierfür geschulte Indianer verantwortlich [Bashshur 80].

**Alaska ATS-6 (1971):** 1971 wurde von der National Library of Medicine das Alaska ATS-6 Satellite Biomedical Demonstration Project initiiert. Innerhalb dieses Projekts wurden 26 Stationen miteinander verbunden. Ziel war es zu untersuchen, ob der Einsatz von Videokonsultationen eine qualitative Verbesserung der medizinischen Versorgung entlegener Gebiete bringen würde. Zu diesem Zweck wurden insgesamt fünf Stationen mit Monochromfernsehern für Videokonferenzen ausgestattet, die eine unidirektionale Video- und eine bidirektionale Sprachkommunikation ermöglichten. Ergebnis dieses Demonstrationsprojektes war, daß Telemedizin bei einer Vielzahl von medizinischen Problemfällen erfolgreich einsetzbar war. Ausgenommen waren allerdings Notfälle, da für solche Fälle die Satellitenverbindung nicht rechtzeitig reserviert werden konnte. Ein deutlicher Vorteil der Videoübertragung gegenüber herkömmlichen Methoden wurde in 5–10% der Fälle konstatiert [Foote 76].

**Ufa (1989):** Das erste internationale Telemedizinprojekt wurde 1989 zwischen den USA und der UdSSR durchgeführt. Nach dem schweren Erd-

beben in Armenien im Dezember 1988 wurde eine Satellitenverbindung zwischen Yerevan und vier Zentren in den USA aufgebaut, um Informationen per Video (unidirektional), Sprache oder Fax zu übertragen. Dieses Projekt demonstrierte die Möglichkeiten der grenzüberschreitenden Telemedizin, in Notfallsituationen schnell Hilfe zu bieten.

## 2.9 Rahmenbedingungen

Die Rahmenbedingungen für den Einsatz der Telemedizin stellen Gesetzgeber, Ökonomie und technische Möglichkeiten dar (Abb. 2.8). Seit der Entwicklung von ISDN, Internet und Webtechnologien sind für viele Bereiche die notwendigen Grundlagen von technischer Seite her geschaffen und die ökonomischen Vorgaben fördern die Telemedizin als Weg, um vorhandenes Personal und Ressourcen effizienter zu nutzen. Die Politik hat mit verschie-

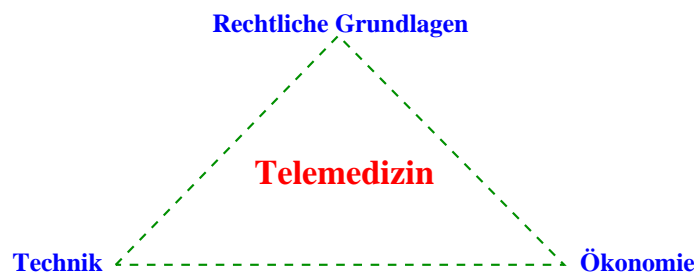


Abbildung 2.8: Die Rahmenbedingungen der Telemedizin bilden technische, rechtliche und ökonomische Möglichkeiten.

denen Gesetzen, u.a. mit dem Signaturgesetz und dem Teledienstgesetz, der Weg für den Einsatz von Telemedizin freigemacht. Eine gute Übersicht zu dieser Thematik bietet die 1997 von der Unternehmensberatung Roland-Berger im Auftrag der Bundesregierung durchgeführte Studie *Telematik im Gesundheitswesen* [RBerger 97].

### 2.9.1 Rechtliche Grundlagen

Beim Einsatz von Telemedizin müssen die dem Arzt bekannten Daten eines Patienten im Rahmen der ärztlichen Schweigepflicht vertraulich behandelt

werden. Die Verarbeitung solcher Daten in EDV-Anlagen regelt das Bundesdatenschutzgesetz (BDSG). In diesem Gesetz wird auch folgende Erläuterung zum Zweck des Datenschutzes gegeben:

*Zweck dieses Gesetzes ist es, den Einzelnen davor zu schützen, daß er durch den Umgang mit seinen personenbezogenen Daten in seinem Persönlichkeitsrecht beeinträchtigt wird.*

Die sich aus diesem Gesetz ergebenden Anforderungen sollen an dieser Stelle nur skizziert werden, Details finden sich in [Ernestus 00b] [Goetz 00a] [Goetz 00b] [Ulsenheimer 00]:

**Vertraulichkeit:** Es muß sichergestellt sein, daß die Daten beim Transport zwischen Sender und Empfänger nicht modifiziert, abgefangen oder abgehört werden können. Technisch erfolgt dies durch Anonymisierung der Daten, Signierung mit sog. digitalen Unterschriften und Verschlüsselung [Ernestus 00a].

**Authentifizierung:** Die Identität eines Benutzers muß bei der Anmeldung an das System überprüft werden. Einfache Authentifizierungsmethoden sind z.B. Kennwortabfrage, sicherere Methoden sind Key Cards oder Biometrische Methoden.

**Verfügbarkeit:** Der Zugriff auf die Daten muß jederzeit gewährleistet sein. Die Daten müssen aktuell sein.

**Minimale Information:** Nur die unbedingt notwendigen Informationen werden übertragen. Es muß sichergestellt sein, daß die Anonymität gewahrt bleibt. Es darf nicht möglich sein, aufgrund der Verkettung der Daten Rückschlüsse auf personenbezogene Daten zu ziehen.

**Selbstbestimmung:** Der Patient muß seine Zustimmung zur Weitergabe seiner Daten geben.

### 2.9.2 Ökonomische Aspekte

Die Motivation bei frühen Telemedizinprojekten (bis 1985) lag in der Verbesserung der Versorgung; seit 1985 und insbesondere seit der Verfügbarkeit von ISDN und Internet werden viele Telemedizin-Systeme aus ökonomischen Gründen installiert. Der Einsatz von Telemedizin erlaubt aus folgenden Gründen eine Kostenersparnis:

**Personal:** Ein kleines Krankenhaus braucht keinen eigenen Spezialisten, sondern nimmt die Dienste eines Spezialisten einer größeren Klinik wahr. Der Spezialist der größeren Klinik wird durch externe Diagnosen besser ausgelastet.

**Kurierdienste:** Die direkte Übermittlung von Laborergebnissen in den OP spart Zeit, der OP wird kürzer belegt.

**Verbesserung Diagnosequalität:** Die Möglichkeit der Konsultation mit einem Kollegen über einen Fall kann die Qualität einer Diagnose erhöhen und so eventuelle spätere Nachbehandlungen, verursacht durch eine falsche Diagnose, verhindern.

### 2.9.3 Technik

Telematikdienste greifen auf Komponenten aus unterschiedlichen Bereichen wie Telekommunikation, Informatik und Anwendungen zurück. Die Integration von neuen Diensten wie der Telemedizin in die IT-Landschaft einer Klinik erfordert eine Verzahnung von existierenden Komponenten mit den neuen.

Die Basis für die Telematik sind der Datentransport über Telekommunikationsnetzwerke, die computergestützte Datenerhebung, digitale bildgebende Systeme und deren Verknüpfung durch Telematikkomponenten (Videokonferenz, Fernsteuerung, Informationsaustausch).

## **2.10 Zusammenfassung**

In Kapitel 2 wurde eine Übersicht über aktuelle Entwicklungen in der Telemedizin, insbesondere der Telediagnostik, gegeben. Es wurden die unterschiedlichen Bereiche der Telemedizin klassifiziert (Telediagnostik, Teletherapie, Telemonitoring, Teleedukation und Teleadministration) und vorgestellt. Besonders betont wurde dabei die Telediagnostik. Aus den Grundlagen zur bildgestützten Diagnostik wurden die Anforderungen an die Telediagnostik abgeleitet und deren Umsetzung in der Teleradiologie und Telepathologie erläutert. Den Schluß des Kapitels bildete ein Überblick über die Rahmenbedingungen und die historischen Entwicklungen der Telemedizin.



# Kapitel 3

## Architektur

Das in dieser Arbeit entwickelte WebTed-System erweitert die Telediagnostik, wie sie mit kommerziell erhältlichen Systemen möglich ist, durch zwei Aspekte: Es macht einerseits als erstes System *quantitative Telepathologie* möglich und ist andererseits komplett webbasiert, d.h. der Anwender benötigt außer einem aktuellen Webbrowser und einem Internetanschluß zur Benutzung keine spezielle Software. Neben der quantitativen Telediagnostik werden auch die statische und die dynamische Telepathologie unterstützt.

Im folgenden werden die Anforderungen an ein System für webgestützte Telepathologie formuliert. Es werden die aktuell verfügbaren Client/Server-Technologien diskutiert und eine Architektur für das Gesamtsystem vorgestellt, welche den Anforderungen genügt.

### 3.1 Anforderungen

Die im vorigen Kapitel vorgestellten Telepathologiesysteme unterstützen nur die statische und die dynamische Telepathologie. Für eine Telediagnose stellen sie Bilder von dünnen Gewebeschnitten bereit, die mit einem Lichtmikroskop bei unterschiedlichen Vergrößerungen aufgenommen wurden. Dazu werden entweder Aufnahmen von Ausschnitten eines Präparats oder das komprimierte Videobild eines ferngesteuerten Mikroskops übertragen. Die übermittelten Bilddaten werden aufgrund von visuellen Merkmalen wie Zell-

form, Zellanordnung und Wachstumsmuster diagnostiziert.

Quantitative Parameter, die die Gewebeänderungen beschreiben, stellen eine wichtige Informationsquelle zur Diagnose und Behandlung von Tumoren dar, werden aber von keinem z.Zt. verfügbaren Telepathologiesystem unterstützt. Der wichtigste Aspekt beim Entwurf des WebTed-Systems war deshalb die Integration einer Bildverarbeitungskomponente in ein Telepathologiesystem.

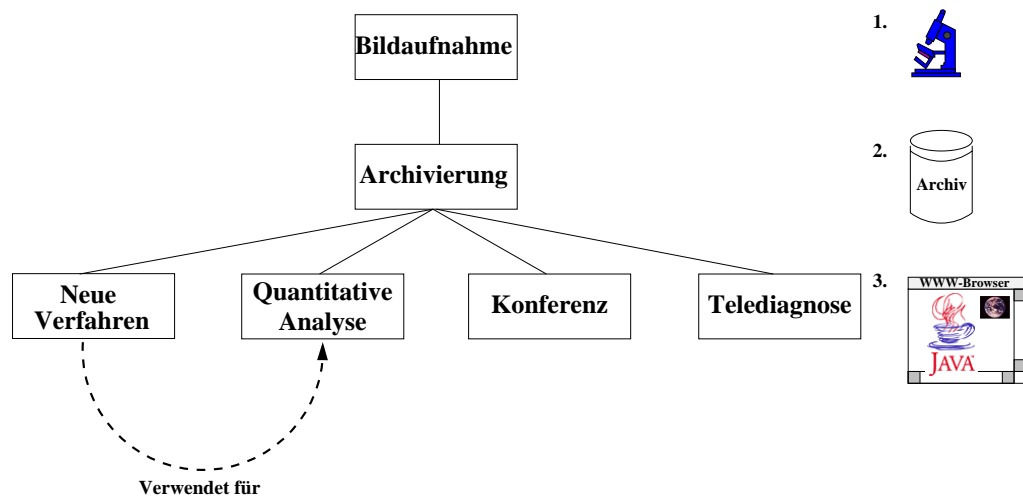


Abbildung 3.1: Workflow für statische Telepathologie/Telequantisierung: Mit einer am Mikroskop angeschlossenen Videokamera werden Aufnahmen von einem histologischen Präparat gemacht und in einem Archiv gespeichert. Sie können dann entweder zur Telediagnostik verwendet oder quantitativ analysiert werden.

Abbildung 3.1 zeigt den Workflow für ein System, das sowohl Telediagnostik als auch Telequantisierung, d.h. die Bestimmung quantitativer Parameter, unterstützt. Der erste Schritt ist die Bildaufnahme. Mit einer an ein Mikroskop angeschlossenen Videokamera werden digitale Bilder aufgenommen und in einer Bilddatenbank archiviert. Die auf diese Weise aufgenommen Bilder werden entweder zur Telediagnostik oder zur Bestimmung quantitativer Parameter verwendet. Die quantitativen Analysemöglichkeiten des Systems lassen sich durch den Entwurf neuer Verfahren erweitern.

Neben der Beurteilung eines Falles durch einen einzelnen Konsultanten ist es auch möglich, einen Fall in einer (Tele)Konferenz zu besprechen, in der

mehrere Betrachter – jeder für sich – das gleiche Bild sehen und Möglichkeiten haben, Regionen im Bild während der Diskussionssitzung zu markieren. Bei diesem Workflow wird von statischer Telepathologie ausgegangen, es läßt sich aber leicht auf dynamische Telepathologie hin erweitern.

Jeder einzelne Schritt des Workflows stellt Anforderungen an das WebTed-System:

**Bildaufnahme:** Mit Hilfe einer graphischen Benutzeroberfläche werden Bilddaten verschiedener Modalitäten aus unterschiedlichen Quellen (Videokamera, Scanner etc.) aufgenommen. Es sollte möglich sein, das System zu einem späteren Zeitpunkt um neue Datenquellen zu erweitern.

**Archivierung:** Die aufgenommenen Bilder werden in einem Archiv gespeichert, um zu einem späteren Zeitpunkt zur Konsultation oder Bildauswertung verwendet zu werden. Der Zugriff muß auf Personen oder Personengruppen einschränkbar sein.

**Telediagnose:** Ein Konsultant hat mit einem Webbrowser auf im System gespeicherte Fälle Zugriff und kann seine Diagnose zu einem Fall mitteilen (statische Telepathologie). Für Konsultationssitzungen, an denen mehrere Konsultanten teilnehmen, wird ein webbasiertes *Whiteboard* benötigt. Es zeigt allen Teilnehmern einer Sitzung das gleiche Bild an; Teilnehmer können dann während einer Diskussion mit Pfeilen auf bestimmte Regionen im Bild hinweisen.

Für die dynamische Telepathologie, die eine freie Auswahl des mikroskopierten Ausschnitts unterstützt, wird ein ferngesteuertes Mikroskop benötigt. Es soll die Kontrolle über die Funktionen Fokus, Position des Präparats, Beleuchtung und Vergrößerung erlauben.

**Quantitative Analyse:** Im Archiv gespeicherte Bilddaten können mit Hilfe von Analyseverfahren, die in einer Methodendatenbank gespeichert sind, ausgewertet werden. Ein Bildverarbeitungsserver führt die Auswertung durch und speichert die Ergebnisse im Archiv. Die Ergebnisse werden dann dem Benutzer in anschaulicher Weise mit Grafiken und

Bildern dargestellt. Der Benutzer soll überprüfen können, ob Verfahrensfehler in der Bildverarbeitung wie z.B. falsch segmentierte Objekte aufgetreten sind.

Zur Erweiterung der Analysefähigkeiten des Systems gibt es eine webgestützte Entwicklungsumgebung zur Implementierung neuer Bildverarbeitungsverfahren. Ist ein Verfahren erprobt, kann es zur allgemeinen Verwendung in eine Methodendatenbank gespeichert werden, um für den Schritt “Quantitative Analyse” eingesetzt zu werden.

Fehlende Kommunikationsstandards in der Telepathologie machen es erforderlich, daß alle Teilnehmer einer Konsultation die gleiche Software verwenden. Dies ist ein Schönheitsfehler im Konzept der Telemedizin: Unabhängig von den lokalen Gegebenheiten (Ort, Ausstattung) soll es möglich sein, einen externen Kollegen um Rat zu fragen. Fehlende Standards schränken die Wahl auf Kollegen ein, die die selbe Software einsetzen.

Das World Wide Web (WWW) bietet einen Ausweg aus diesem Dilemma: Über das WWW können Informationen mit einem Webbrowser von einem Server abgerufen werden. Webbrowser sind für alle in der Praxis relevanten Plattformen verfügbar (Windows, Linux/Unix, Macintosh). Damit gibt sich die Möglichkeit, die für eine Konsultation notwendigen Daten (Texte, Bilder) auf einem Server bereitzuhalten. Aufgrund der hohen Anforderungen des Datenschutzes werden die personenbezogenen Daten in einem geschützten Bereich auf dem Server gehalten. Konsultanten rufen sie dann unter Angabe von Benutzername und Kennwort vom Server ab, um die Daten in ihrem Webbrowser anzuzeigen.

Das WWW war ursprünglich als Informationssystem für die Wissenschaft gedacht. Zahlreiche Erweiterungen (Java, JavaScript, Plugin-Konzepte, dynamische Seitengenerierung durch den Webserver) erlauben inzwischen eine vielseitige Verwendung des WWW. Ein Überblick über die Technologien findet sich in Abschnitt 3.3.

### 3.1.1 Anwendungsbereich

Anwender für die Telediagnostik ist die Arbeitsgemeinschaft Knochentumoren (AG-KT), die am DKFZ angesiedelt ist. Die AG-KT ist ein internationales, interdisziplinäres Gremium zur Erforschung der Diagnostik und Therapie von Knochentumoren. Ihr gehören Pathologen, Orthopäden, Radiologen sowie Grundlagenforscher an, die sich zweimal pro Jahr treffen, um über schwierige Fälle zu beraten.

In den 26 Jahren der Tätigkeit der AG-KT wurden etwa 700 Fälle diskutiert. Für die Diskussion eines Falles auf einem Treffen wurde bisher für jedes Mitglied von der Gewebeprobe ein separates Präparat angefertigt und zusammen mit den Dias der Röntgenbilder verschickt. Jedes Mitglied erstellte dann eine eigene Diagnose. Auf einem Treffen wurden dann die unterschiedlichen Diagnosen der Mitglieder diskutiert und eine gemeinsame Diagnose erstellt.

Die Arbeitsgemeinschaft setzt Telediagnostik ein, um schneller als bisher von Experten aus unterschiedlichen Fachgebieten (Pathologen, Radiologen, Orthopäden) eine Expertise einholen zu können. Dabei wird die statische Telepathologie aus zwei Gründen eingesetzt:

1. Eine zeitliche Koordinierung einer Konsultation, wie sie für die dynamische Telepathologie notwendig ist, ist nur in den wenigsten Fällen möglich. Die statische Telepathologie erlaubt es dem Konsultanten, sich dann den Fall anzusehen, wenn er Zeit dafür hat. Es ist keine vorherige Absprache notwendig.
2. Für Expertenkonsultationen ist die statische Telepathologie ausreichend (siehe Seite 12). Sie läßt sich bei einem minimalen Einsatz an Ressourcen (Bandbreite des Netzwerkes, Rechenpower, benötigt kein ferngesteuertes Mikroskop) gut einsetzen. Es wird für eine Diagnose weniger Zeit benötigt, als beim Einsatz von dynamischer Telepathologie, die Erstellung einer Diagnose mittels Telemikroskopie dauert etwa doppelt so lange, wie wenn die Diagnose lokal unter dem Mikroskop erstellt wird.

Ein weiterer Vorteil der Telediagnostik gegenüber der konventionellen Dia-

agnostik ist die digitale Form der Bilddaten. Bei der Anfertigung von bis zu 40 Schnitten pro Tumor kann es vorkommen, daß einzelne Mitglieder der AG unterschiedliche Regionen der Gewebeproben beurteilen. Außerdem weisen gefärbte histologische Präparate eine unterschiedliche Qualität auf, die zu unterschiedlichen Diagnosen führen kann. Diese Art von Artefakten kann durch den Einsatz von Telediagnostik vermieden werden.

Als erster Schritt wurde innerhalb der AG-KT webbasierte Telediagnostik eingeführt. Als nächster Schritt soll die quantitative Telepathologie zur Unterstützung von Pathologen bei der Diagnose eingeführt werden. Es werden Verfahren eingesetzt, die in verschiedenen Diplom- und Doktorarbeiten entwickelt wurden. Ein Teil dieser Verfahren ist bereits in das System integriert worden, sie werden intern zur Unterstützung bei der Diagnose eingesetzt. Diese Verfahren wurden jeweils nur an kleinen Datenmengen evaluiert, so daß es vor dem allgemeinen Einsatz der quantitativen Telepathologie notwendig ist, die Verfahren anhand von großen Datenmengen zu verifizieren.

## 3.2 Lösungsansatz

Die Telediagnostik erfordert die Erhebung von Daten (Bilder, klinische Daten) und deren Beurteilung an verschiedenen Orten, es ist also notwendig, die Daten zur Beurteilung über ein Netzwerk zu verschicken. Der sinnvollste Ansatz für eine solche Konstellation ist eine Client/Server-Architektur, bei der die Daten an verschiedenen Punkten erhoben und über ein Netzwerk auf einem zentralen Server gespeichert werden. Von dort aus lassen sie sich von verschiedenen Stationen aus zur Erstellung einer Diagnose betrachten.

In der Informatik bezeichnet Client/Server-Architektur die Beziehung zwischen Prozessen, die auf unterschiedlichen Maschinen laufen. Der Server stellt dabei unterschiedliche Dienste bereit, auf die ein Benutzer über eine Client-Software zugreifen kann [Orfali 96]. Die wichtigsten Vorteile einer Client/Server-Architektur sind:

**Plattformübergreifend:** Client und Server müssen nicht das gleiche Betriebssystem verwenden, es muß lediglich eine Software existieren, die

den Zugriff auf die Serverdienste erlaubt. Client und Server sind über ein Netzwerk miteinander verbunden.

**Ressourcen:** Der Server kann Ressourcen (z.B. Datenbanken, Dateien) zentral bereithalten, auf die dann die Clients über ein Netzwerk zugreifen. Die Verwaltung der Daten gestaltet sich dadurch einfacher.

**Verwaltung:** Daten werden an nur einer zentralen Stelle, dem Server, gehalten. Nur der Server muß mit hoher Sorgfalt gewartet werden; dies impliziert einen geringeren Wartungsaufwand für die Client Rechner. Bei sensiblen Daten muß nur die Sicherheit auf dem Server überwacht werden.

Eine der bekanntesten Client/Server Implementierungen ist das WWW. Mit einem Webbrowser lassen sich unterschiedliche Dienste nutzen, die von vielen Webservern bereitgehalten werden. Die Vorteile liegen auf der Hand: Auf den meisten Rechnern ist heutzutage ein Webbrowser installiert, so daß der gleiche Client für viele Dienste verwendet werden kann. Im folgenden werden die heute verfügbaren Technologien und ihre Anwendungsbereiche vorgestellt, ein Überblick über die Architektur des WebTed-Systems gegeben und die Wahl der verwendeten Technologien begründet.

### 3.3 Client/Server im Web

1993 wurde das World Wide Web (WWW) als ein universelles, plattformübergreifendes Client/Server-Informationssystem zum Abrufen von Dokumenten von Servern eingeführt. Bei den Dokumenten handelt es sich um sog. Hypertext-Dokumente, die neben Text auch Verweise auf Bilder und andere Dokumente enthalten. Mit einem universellen Client, dem Webbrowser, werden Dokumente, die Texte und Bilder enthalten, von einem Web-Server abgerufen und dargestellt. Client und Server kommunizieren mittels HTTP-Protokoll (HyperText Transfer Protocol, RFC 2616), einem einfachen textorientiertem Protokoll, miteinander. Der Aufbau der Dokumente ist in HTML (HyperText Markup Language, RFC 1866) beschrieben. HTML ist

eine einfache Seitenbeschreibungssprache die Texte, Bilder, Tabellen, eingebettete Objekte (Java Applets, Plugins, ActiveX Elemente) und Verweise auf andere Dokumente in einem Textdokument beschreibt [W3C].

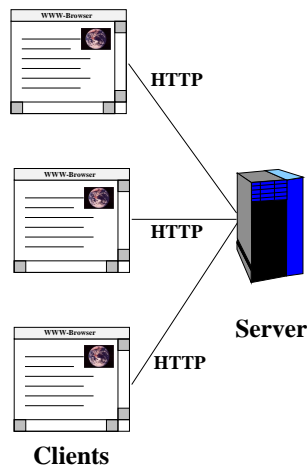


Abbildung 3.2: *Das WWW als Client/Server Architektur. Mit dem Webbrowser werden Dokumente vom Server abgerufen und dargestellt.*

Ursprünglich wurde das WWW nur zum Abrufen von Dokumenten entwickelt; ab 1995 wurden durch Erweiterungen wie FORMS in HTML 2.0 und die dynamische Erzeugung von Webseiten auf dem Server mit cgi-bin Anwendungen interaktive Applikationen ermöglicht. Ab 1996 erlaubte es die Integration von Java-Interpretern in gängige Webbrowser, Programme von einem Server zu laden und auf dem Client ablaufen zu lassen. Die technische Entwicklung des WWW ist in Abbildung 3.3 skizziert.

Es gibt zwei prinzipielle Methoden, um innerhalb des WWW interaktive Applikationen zu erstellen. Sie sind ein wichtiger Aspekt bei der Architektur des WebTed-Systems und werden deshalb kurz vorgestellt. Die eine Methode besteht darin, den Großteil der Funktionalität auf dem Server ablaufen zu lassen (cgi-bin) und die Ergebnisse im Browser darzustellen; bei der zweiten Methode werden in eine HTML-Seite integrierte Java Applets verwendet. Java Applets laufen auf der Client-Seite, sie können Daten mit dem Server austauschen und im Browser darstellen. Beide Verfahren haben Vor- und Nachteile, sie werden in den folgenden Abschnitten diskutiert.



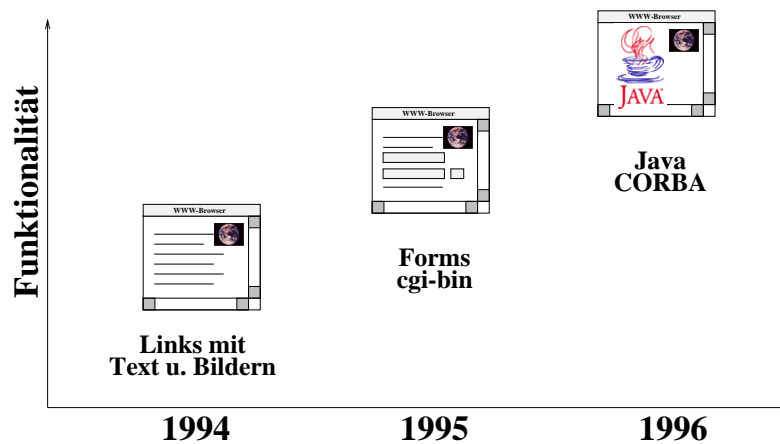


Abbildung 3.3: Entwicklung der WWW Funktionalität: 1994 unterstützte das Web nur die Darstellung von Hypertextdokumenten. Durch die Einführung von Formularen (Forms in HTML 2.0) wurden einfache Anwendungen möglich. Erst die Integration von Java in Webbrowser erlaubte hochinteraktive Anwendungen.

### 3.3.1 Serverseitige Informationsverarbeitung: cgi-bin

Anwendungen, die mit einer niedrigen Interaktivität auskommen, lassen sich durch serverseitig dynamisch generierte HTML-Dokumente realisieren. Der Benutzer ruft vom Server ein HTML-Dokument ab. Auf dieser Seite werden Daten in Formularfelder (HTML Forms) eingegeben und an den Webserver geschickt. Der Server startet daraufhin ein Programm, das die Eingaben auswertet, eventuell Daten aus einer Datenbank abfragt, und als Ergebnis eine HTML-Seite generiert, die an den Browser geschickt wird. Die Daten des Webformulars werden dem cgi-bin Programm dabei über Umgebungsvariablen und Standardinput übergeben, das Programm schreibt die HTML-Seite auf Standardoutput. Die Schnittstelle zwischen Webserver und diesem Programm wird als cgi bezeichnet (Common Gateway Interface).

In den meisten Fällen werden Skriptsprachen (Perl, Python, Netscape JavaScript, Microsoft Active Server Pages) für cgi-bin Programme verwendet. Die Entwicklung einer cgi-bin Anwendung erfordert drei Schritte. In einem ersten Schritt wird das HTML-Formular entworfen, anschließend muß das cgi-bin Programm zum Verarbeiten der Daten entwickelt werden. Zum

Schluß muß dann die Seite entworfen werden, die als Ergebnis an den Browser geschickt wird.

Die Geschwindigkeit von cgi-bin Anwendungen ist beschränkt durch die benötigte Ausführzeit für die cgi-bin Anwendung, die vom Browser benötigten Zeit für die Darstellung dieser Seite und die für den Aufbau einer Netzverbindung zum Server benötigten Zeit. Die Interaktivität bei cgi-bin Anwendungen liegt typischerweise im Sekundenbereich und ist für viele Anwendungen ausreichend.

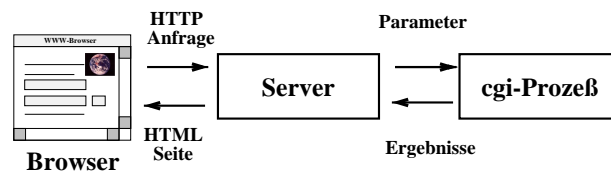


Abbildung 3.4: *Webserver mit cgi-bin Erweiterung: Der Webbrowser sendet eine Anfrage an den Server. Der Server übergibt die Parameter als Umgebungsvariablen einem cgi-bin Programm. Das cgi-bin Programm erzeugt dynamisch eine HTML Seite, die vom Server an den Webbrowser geschickt wird.*

### 3.3.2 Clientseitige Informationsverarbeitung: Java

Hochinteraktive Anwendungen lassen sich mit Java entwickeln. Java ist eine objektorientierte Programmiersprache, die speziell unter den Aspekten Plattformunabhängigkeit und Sicherheit für mobilen Code von Sun Microsystems entwickelt wurde. Es wird nicht – wie bei kompilierten Hochsprachen wie C++ – Maschinencode für einen *bestimmten* Prozessortyp, sondern für einen *abstrakten* Java Prozessor generiert. Dieser aus dem Java Sourcecode erstellte Maschinencode wird als “Bytecode” bezeichnet. Um den Code ausführen zu können, wird eine Emulation des abstrakten Prozessors benötigt, sie wird durch die Java Virtual Machine (JVM) bereitgestellt.

Eine der großen Stärken von Java ist die integrierte Sicherheitsarchitektur. Java Programme haben keinen direkten Zugriff auf das Betriebssystem eines Rechners, sie können lediglich auf die Funktionen der JVM zugreifen

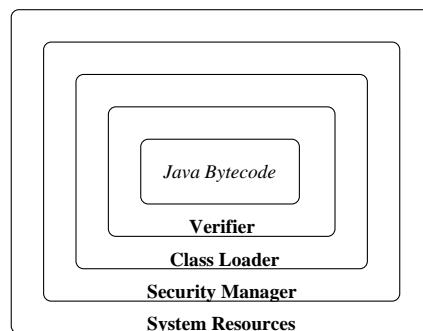


Abbildung 3.5: *Die Java Sicherheitsarchitektur: Vor der Ausführung von Bytecode wird dieser zunächst vom Verifier auf Richtigkeit hin überprüft und vom Class Loader in die JVM geladen. Der Class Loader überwacht den Ursprung (Netz, Datei) der Klassen. Zugriffe auf Systemressourcen werden vom Security Manager auf die Berechtigung überprüft und dann freigegeben oder abgelehnt.*

(Abb. 3.5). Die JVM läßt allerdings nur Operationen zu, die entweder unbedenklich sind, oder aber durch den Benutzer explizit autorisiert wurden (*Sandbox-Architektur*). Die Details sollen an dieser Stelle nicht erklärt werden; ausführliche Informationen zu Java finden sich z.B. im Java Tutorial oder direkt bei Sun ([HTTP://JAVA.SUN.COM](http://java.sun.com)) [Java Tutorial].

Alle modernen Browser (z.B. Netscape Communicator und Microsoft Internet Explorer) enthalten mittlerweile eine JVM. Damit können Java-Komponenten, die sogenannten Java Applets, direkt in eine Webseite integriert werden. Der Java Bytecode für das Applet wird vom gleichen Webserver wie die HTML-Seite geladen. Das Java Applet stellt eine grafische Benutzeroberfläche bereit, um Daten und Bilder darzustellen und kann mit dem Server kommunizieren und Eingaben an den Server schicken.

Im Gegensatz zu den serverseitig verarbeiteten cgi-bin Anwendungen findet bei Java Applets der Großteil der Informationsverarbeitung clientseitig statt. Daten lassen sich auf der Client-Seite verarbeiten, es muß nicht wie bei cgi-bin basierten Anwendungen der Umweg über den Server gemacht werden. Dadurch werden höhere Interaktionsraten erreicht.

Die Ausführungszeit von Java ist durch die Emulation prinzipbedingt erheblich langsamer als bei kompilierten Sprachen wie C++, ist aber

auf modernen PCs für die meisten Anwendungen ausreichend. An der Steigerung der Performance von Java Anwendungen durch Java-Chips, Just In Time Compiler, Adaptive Compiler und Native Compiler wird gearbeitet, eine Übersicht findet sich in [Byte 6/98]. An dieser Stelle soll auf die Diskussion darüber verzichtet werden, vielmehr soll auf die Möglichkeit der Client/Server Programmierung mit CORBA und Java Remote Method Invocation (RMI) eingegangen werden.

### **Verteilte Objekte: CORBA, Java Remote Method Invocation**

Die Kommunikation zwischen Client und Server erfordert ein Protokoll, das den Informationsaustausch regelt. Ein Beispiel für ein solches Protokoll ist HTTP für die Kommunikation zwischen Webbrowser und Webserver.

Entwurf und Implementierung von Protokollen sind zeitaufwendig und fehlerträchtig. Um diesen Aufwand zu umgehen, wurden schon sehr früh Verfahren entwickelt, um Protokolle automatisch zu generieren. Solche Methoden sind z.B. Remote Procedure Call (RPC, RFC 1057) unter Unix, um Funktionen auf entfernten (Unix) Rechnern aufzurufen, oder aber CORBA und Java Remote Method Invocation (RMI) als objektorientierte Varianten. RPC wird im Unix Bereich eingesetzt, um zwischen Unix Rechnern zu kommunizieren, z.B. für die Implementierung von NFS (Network File System). Die Kommunikation mit RPC ist lediglich zwischen Programmen möglich, die mit der Programmiersprache C entwickelt wurden, eine Kommunikation zwischen Java und C ist über RPC nicht möglich. RPC ist deshalb für die Verwendung in Client/Server Webanwendungen nicht ungeeignet und wird hier nicht weiter diskutiert.

Verfahren wie RPC, CORBA und RMI basieren auf dem Prinzip der Trennung von Schnittstelle und Implementierung (Abb. 3.6). In einem Interface wird die Schnittstelle zu den Serverdiensten definiert, das dann einmal für den Server und als Proxy-Objekt für den Client implementiert wird (vergl. Proxymuster in [Gamma 96]). Die Erzeugung des Codes für den Client-Proxy erfolgt automatisch durch den Interface-Compiler. Der Client-Proxy enthält den Code mithilfe dessen Aufrufe für Methoden des Interfaces über

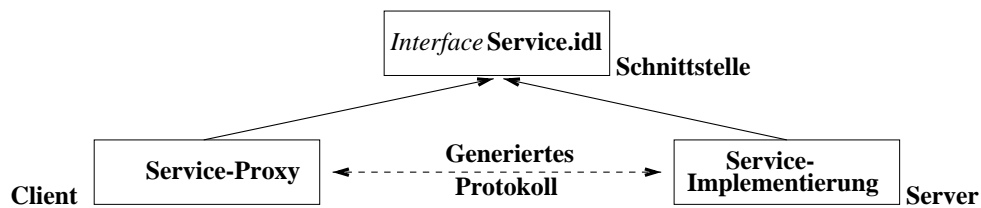


Abbildung 3.6: Die Dienste eines CORBA oder Java Servers werden in einer Interface definiert. Die abstrakte Interfaceklasse wird einmal als Server und einmal als Service-Proxy implementiert. Beide Implementierungen enthalten identische Methoden, aber die Proxy-Implementierung leitet alle Anfragen über ein Netzwerk an die Server Implementierung weiter.

ein Netzwerk an den Server weitergeleitet werden.

Ein Client verwendet das Proxy-Objekt, um Aufrufe beim Server durchzuführen. Die Netzwerkkommunikation ist dadurch für den Entwickler transparent und ermöglicht einfache Erweiterungen der Serverdienste ohne manuelle Änderungen des Protokolls. Das hier beschriebene Verfahren wird von RPC, CORBA und Java RMI verwendet.

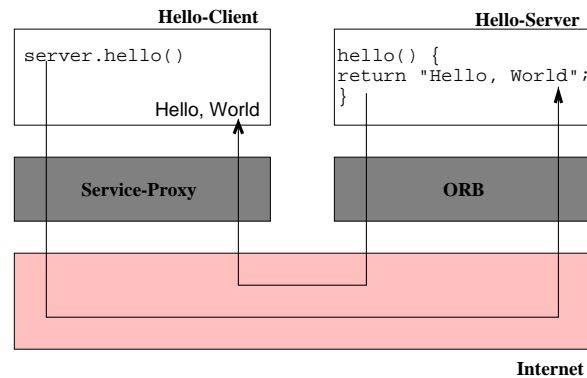


Abbildung 3.7: Hello World als Client/Server Programm mit CORBA: Der Client ruft die Servermethode `hello()` auf. Die `hello()` Methode des Service-Proxy ruft über den Object Request Broker (ORB) die `hello()` Methode des Servers aus und liefert den String "Hello, World" zurück.

CORBA ist eine universelle Middleware<sup>1</sup> für die Kommunikation zwischen beliebigen Objekten. Sie wurde von einem Konsortium, bestehend aus 800

<sup>1</sup>Als Middleware wird eine Kommunikationssoftware bezeichnet, die Zugriff den auf Softwaremodule über Netz- und Systemgrenzen hinweg erlaubt.

Firmen aus dem IT-Sektor, der Object Management Group, formuliert, um einen Standard für verteilte objektorientierte Systeme zu schaffen [OMG]. CORBA erlaubt die Interoperabilität unabhängig von:

- der Programmiersprache, in der Objekte implementiert wurden.
- dem Ort, an dem sie sich im Netzwerk befinden.
- der CORBA Implementierung, die verwendet wurde.

Für die Implementierung wird das Objekt mit der CORBA eigenen Interface Definition Language (IDL) beschrieben. Der Interface-Compiler generiert dann für die Zielsprache (C/C++, Fortran, Cobol, Java) den Sourcecode für die Kommunikation. Damit ein Client Methoden eines Serverobjektes aufrufen kann, wird als erstes eine Referenz auf das Serverobjekt benötigt. Jedes Serverobjekt meldet sich beim Object Request Broker an (ORB). Der Client erhält vom Naming Service des ORBs eine Referenz auf das Serverobjekt, Methodenaufrufe werden dann über den ORB an das entsprechende Objekt weitergeleitet und ausgeführt. Weitere Informationen zu CORBA finden sich in [Redlich 96].

CORBA läßt sich auch für die Objektkommunikation zwischen Java/Java und Java/C++ verwenden. Gegen die Verwendung von CORBA spricht die Größe von CORBA. Serverseitig ist dies unproblematisch, wird aber ein Java Applet als Client eingesetzt, müssen allein für CORBA etwa 400 KB an Klassen übertragen werden. Dies ist für ein Applet bei einer langsamen Modemverbindung sehr viel, eine Alternative hierzu ist Java RMI.

Java RMI ist sehr ähnlich zu CORBA, ist aber nicht programmiersprachenübergreifend. Es läßt sich nur für die Verteilung von Java-Objekten innerhalb eines Netzwerkes verwenden.<sup>2</sup> Für RMI sind alle erforderlichen Klassen in der JVM integriert, es muß nur der Client-Proxy für den Zugriff auf den Server übertragen werden. Dieser hat typischerweise eine Größe zwischen 10 und 20 KB.

---

<sup>2</sup>Ab Java 2 verwendet Java RMI optional das CORBA IIOP Protokoll und kann auf Objekte zugreifen, die durch ein CORBA ORB bereitgestellt werden [RMI IIOP].

Mit CORBA oder Java RMI lassen sich Anwendungen erheblich schneller entwickeln und erweitern; es läßt sich natürlich nicht dieselbe Effizienz erreichen wie mit manuell erstellten, optimierten Protokollen. Die benötigte Zeit, um eine Methode eines Objekts ohne Argumente und Rückgabewerte aufzurufen, beträgt bei einer Verteilung der Objekte in einem lokalen Netzwerk mit Java RMI auf einem Pentium 200 unter Linux im LAN etwa 3 ms. Für die Telepathologie hat sich die Performance von Java RMI als ausreichend bestätigt.

### 3.4 Zwischenergebnis

Insgesamt gesehen läßt sich sagen, daß das World Wide Web eine elegante Möglichkeit für die Implementierung von Client/Server-Systemen bietet. Ein universeller Client, der Webbrowser, läßt sich für ein weites Anwendungsspektrum einsetzen. Außer dem Webbrowser ist keine weitere vorinstallierte Software auf dem Client Rechner notwendig. Die Informationsverarbeitung kann entweder überwiegend auf dem Server (cgi-bin), oder aber auf dem Client (Java) stattfinden. Die erste Möglichkeit läßt eine Interaktivität von einigen Sekunden zu, die zweite ist etwas langsamer als gewöhnliche Client Applikationen, die mit C++ erstellt wurden. Mit verteilten Objekttechnologien wie CORBA oder Java RMI existiert eine effiziente Methode der Client/Server Programmierung: Objekte auf der Serverseite können vom Client aus angesprochen werden, als seien sie lokal auf dem Client vorhanden. Dies erspart die Implementierung von speziellen Protokollen für die Client/Server Kommunikation.

Nicht diskutiert wurden an dieser Stelle proprietäre Erweiterungen des Browsers wie ActiveX-Komponenten (Microsoft) oder Communicator Plugins (Netscape). Sie erweitern die Funktionalität des Webbrowsers und können ähnlich wie Java Applets in HTML-Seiten integriert werden, allerdings müssen sie speziell installiert werden (Netscape Plugin) und sind als unsicher einzustufen, da sie im Gegensatz zu Java Applets Zugriff auf alle Funktionen des Betriebssystems haben. Es ist deshalb möglich, mit einem Plugin-Dateien

zu lesen, zu schreiben und zu löschen, beliebige Netzverbindungen zu anderen Computern aufzubauen etc.. Ebenfalls nicht diskutiert wurde Netscape JavaScript. JavaScript ist eine Skriptsprache, die clientseitig eingesetzt wird. Mit ihr lassen sich einfache Funktionen (Fenster mit einer URL öffnen, Fenster schließen, Menüleiste verstecken, Werte in einem Formular überprüfen, Dialog anzeigen etc.) im Browser in einem Skript aufrufen. JavaScript ist für einfache Funktionalitäten auf der Client-Seite gut geeignet, für größere Anwendungen aber ungeeignet. Es kann nicht verwendet werden, um mit CORBA oder RMI auf verteilte Objekte zuzugreifen.

Nicht verschwiegen werden soll an dieser Stelle, daß es in der Vergangenheit Sicherheitslücken in der JVM gegeben hat. Grund waren Schwächen in der Implementierung der JVM, die schnell behoben wurden und nicht in der Architektur der JVM begründet waren [McGraw 96] [McGraw 99]. Nichtsdestoweniger bietet Java von allen Konzepten für aktive Inhalte (JavaScript, Netscape Plugins, ActiveX) laut einer Studie des Bundesamtes für Sicherheit in der Informationsverarbeitung z.Zt. die größte Sicherheit im Internet [OCOCAT 99].

### 3.5 Implementierung

In den vorigen Abschnitten wurde die Verwendung von Java aufgrund der integrierten Sicherheitsarchitektur, des plattformunabhängigen Ansatzes und der Integration von Java Applets in Web-Seiten motiviert.

Ein Java Applet dient als grafische Benutzerschnittstelle zu den Serverdiensten statische, dynamische und quantitative Telepathologie. Jeder dieser Dienste stellt Anforderungen an die Client Software. Die Anforderungen sind im einzelnen:

**Akquisition von Bilddaten:** Um für die statische Telepathologie Bilddaten auf dem Server ablegen zu können, müssen diese vom Java Applet vom Computer des Benutzers gelesen werden, entweder als Datei oder direkt aus einer Framegrabberkarte. Es wird also Zugriff auf das Dateisystem bzw. auf eine Shared Library für die Ansteuerung des Frame-



grabbers benötigt.

Die notwendigen Privilegien um Dateien zu lesen bzw. Funktionen der Shared Library aufzurufen, kann der Benutzer einem Applet einräumen, wenn es mit einer digitalen Unterschrift signiert ist (*Object Signing*) [Gong 99]. Damit wird sichergestellt, daß das Applet während des Transports zwischen Server und Client nicht modifiziert wurde und es ist auch der Urheber des Applets sichergestellt.<sup>3</sup>

**Netzwerkzugriff:** Die Software für die dynamische Telepathologie läuft auf einem anderen Server als die übrige Software, es ist daher erforderlich auch zu diesem Server Netzwerkverbindungen aufzubauen. Ohne besondere Privilegien erlaubt die JVM nur Netzwerkverbindungen zu dem Rechner, von dem das Java Applet geladen wurde. Mit Object Signing kann der Benutzer dem Applet erlauben, auch andere Netzwerkverbindungen herzustellen.

Besondere Privilegien sind nur für dynamische Telepathologie (Netzwerkzugriff) und statische Telepathologie zum Ablegen von neuen Fällen (Zugriff auf Datei/Framegrabber) notwendig. Zum Betrachten von Fällen der statischen Telepathologie oder für die Verwendung des Bildverarbeitungsmoduls werden keine besonderen Privilegien benötigt. Die Details zu den Privilegien werden in Abschnitt 3.5.3 erläutert.

### 3.5.1 Benutzeroberflächen

Es gibt unter Java zwei Alternativen für die Implementierung grafischer Benutzeroberflächen, nämlich das AWT (Abstract Window Toolkit) und die neueren Java Foundation Classes (JFC, auch Swing genannt). Das AWT verwendet für jede grafische Komponente wie Button, Listen etc. die entsprechenden Komponenten des Betriebssystems. Das Verfahren ist in Abb. 3.8

---

<sup>3</sup>Eine Alternative hierzu wäre das FILE Tag in einem HTML Formular gewesen (RFC 1867). Es erlaubt die Auswahl *einer einzelnen* Datei über einen Filedialog und den Versand dieser Datei mit POST an ein cgi-bin Programm. Die Java Alternative wurde implementiert, da damit *ganze Verzeichnisse* mit Bilddaten versendet werden oder Bilddaten direkt aus der Framegrabberkarte ausgelesen werden können.

dargestellt. Um eine Instanz eines Buttons zu erzeugen, wird von der Java

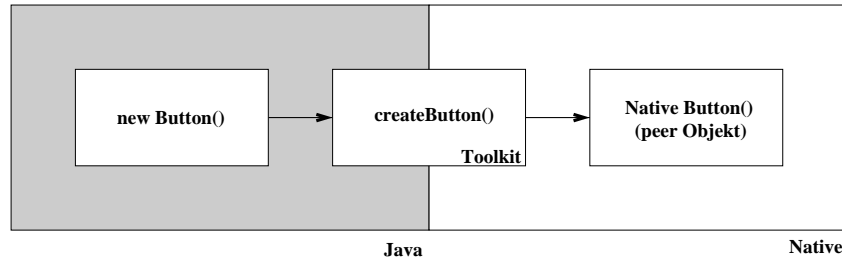


Abbildung 3.8: Um ein grafisches Objekt wie einen Button zu instanzieren, ruft die Button Klasse die Methode `Toolkit.createButton()` auf, die dann beim Betriebssystem den Aufruf zum Erzeugen eines Buttons durchführt.

Toolkit Implementierung für das jeweilige Betriebssystem eine sog. Peer-Funktion aufgerufen, die beim Betriebssystem einen Aufruf zur Erzeugung eines Buttons durchführt. Dies führt dazu, daß ein Java Programm das “Look and Feel” der Plattform besitzt, auf der es gerade läuft. Ein solches Vorgehen hat aber auch Nachteile: Für die Implementierung der JVM muß für jedes Betriebssystem ein Toolkit implementiert werden, und nicht alle Komponenten sind auf allen Betriebssystemen vorhanden. Das AWT ist deshalb der kleinste gemeinsame Nenner. Das Portieren der Netscape JVM auf 17 Plattformen wurde von Netscape als zu arbeitsintensiv angesehen, weshalb Netscape die Netscape Foundation Classes entwickelte, aus denen dann in Kooperation mit Sun die Java Foundation Classes (JFC) wurden.<sup>4</sup>

Die JFC sind der Standard für Benutzerschnittstellen unter Java 2.0. Sie verwenden zum großen Teil Light-Weight-Komponenten, d.h. grafische Komponenten wie Buttons etc. werden nicht mehr vom Betriebssystem gezeichnet, sondern direkt von der Java Klasse. Für eine Portierung der JVM fällt deshalb ein Großteil der Peer-Funktionen weg, es muß nur die Basisfunktionalität zum Zeichnen von Linien, Fenstern etc. portiert werden. Das Aussehen einer Applikation wird deshalb nicht mehr vom verwendeten Betriebssystem festgelegt, sondern vom eingestellten Look-and-Feel: Anstelle einer Implementierung des AWT für jedes Betriebssystem gibt es für jede

<sup>4</sup>Interessanterweise ist die Entscheidung für ein plattformspezifisches Aussehen ursprünglich von Netscape forciert worden [Gosling 98]. Ursprünglich war für Java 1.0 kein plattformspezifisches Look-and-Feel vorgesehen.

grafische Komponente eine Objektfabrik. Je nach eingestelltem Look-and-Feel wird eine Komponente mit Windows, Macintosh, CDE/Motif und Java Cross-Plattform Look-and-Feel erzeugt.

Die JFC können mit dem aktuellen Stand von Windows-Benutzeroberflächen konkurrieren, sie enthalten Komponenten wie Buttons mit Bildern, Multiple Document Interface Fenster (MDI), Drag-and-Drop, aktive Hilfehinweise mit einer ähnlichen Funktionalität wie Sprechblasen unter Mac OS sowie Widgets zum Anzeigen von HTML-Texten.

### 3.5.2 JVM in der Praxis

So schön Java, JFC und Plattformunabhängigkeit in der Theorie auch sein mögen, so treten in der Praxis doch Probleme auf. Für Java wurden viele Pakete für Datenbankanbindungen, IDE, Compiler, Debugger, CASE Tools, Komponentenmodelle und Bibliotheken für 3D-Grafik und Bildverarbeitung in kurzer Zeit neu entwickelt; dabei konnte wegen der Plattformunabhängigkeit jedoch nur in wenigen Fällen auf existierende C/C++ Software zurückgegriffen werden. Gleichzeitig entwickelte sich Java sehr schnell weiter, so daß die Qualität von Java Produkten nicht immer dieselbe Qualität wie Produkte aufwies, die für C/C++ entwickelt wurden und für die schon eine lange Erfahrung existiert.

Dies trifft insbesondere auch für die JVMs in den Browsern zu; vor allem die Implementierung der AWT hat in älteren Browsern viele Fehler gehabt, die es zu umschiffen galt. Die für die Entwicklung von Anwendungen verfügbaren JVMs bereiten im Gegensatz zu den JVMs in Browsern nur selten Probleme, sie basieren zum großen Teil auf der Original JVM von Sun. Die JVMs in den Browsern sind um etwa ein Jahr der aktuellen Entwicklung von Sun hinterher, Microsoft integriert aus politischen Gründen (DCOM) nicht alle Standard Klassen in ihre JVM.<sup>5</sup> Die Probleme mit den JVMs sind bekannt, eine Lösung hierzu ist das Open Java Interface, eine Schnittstelle, um eine beliebige JVM in den Browser zu integrieren. Die Open Java

---

<sup>5</sup>Es fehlen der Microsoft Implementierung die Klassen für Java Remote Method Invocation (RMI); die RMI Klassen gehören zum Kern-Package (java.\*) und können deshalb aus Sicherheitsgründen auch nicht einfach vom Web-Server geladen werden.

Schnittstelle wird im Netscape Communicator 6.0 integriert sein und dürfte die meisten Probleme mit den JVMs beseitigen [OJI].<sup>6</sup>

Leider sind die Klassen für JFC noch nicht in den aktuellen Browsern vorhanden; ein Applet, das die JFC verwendet, muß deshalb die JFC Klassen vom Webserver laden. Dies wäre an sich kein Problem, wenn die Klassen zusammen nicht über 2 MB groß wären. Auch bei einer schnellen Netz-anbindung, wie sie z.B. innerhalb des Deutschen Forschungsnetzes gegeben ist, dauert das Laden eines solchen Applets relativ lange: Alle Klassen, die vom Server geladen werden, müssen durch die JVM verifiziert werden. Zwar wäre es möglich, die JFC Klassen lokal auf dem Client zu installieren, das widerspräche aber der Philosophie der Java Applets.

Auch Sun hat dieses Dilemma erkannt und bietet deshalb ein Java-Plugin für Microsoft IE und Netscape Communicator an. Das Java Plugin verwendet die neueste Version der JVM; Applets die mit dem HTML-EMBED-Tag in eine Webseite integriert werden, verwenden dann die aktuelle JVM von Sun anstelle der von Netscape oder Microsoft. In die Plugins ab Version 1.1.3 sind die JFC Klassen integriert, so daß diese nicht über das Netzwerk geladen werden müssen. Für das Applet wird deshalb das Java-Plugin von Sun verwendet; das Applet verlangt zwar eine Installation des Plugins, erlaubt dem Benutzer aber ein schnelleres Laden der Applets. In der nächsten Browser Generation werden die JFC Klassen integriert sein, und es kann eine beliebige JVM verwendet werden; bis dahin muß jedoch abgewartet werden.

### 3.5.3 Privilegien

Ein Teil der Anwendungen für die Telepathologie benötigt besondere Privilegien wie z.B. Zugriff auf lokale Dateien, solche Privilegien können vom Benutzer unter bestimmten Bedingungen an das Applet gewährt werden. Sun, Netscape und Microsoft verfolgen dabei eine ähnliche Strategie, sind aber nicht kompatibel zueinander. Damit ein Benutzer einem Java Applet mit gutem Gewissen zusätzliche Privilegien einräumen kann, muß sichergestellt sein, *von wem das Applet kam* und daß das *Applet nicht bei der*

---

<sup>6</sup>Der Communicator 6.0 wird im 2. Quartal 2000 verfügbar sein.

*Übertragung modifiziert* wurde. In der Kryptographie wird dies durch eine digitale Unterschrift erreicht. Digitale Unterschriften basieren auf einem Public-Key Kryptographieverfahren [Schneider 99]. Public-Key Verfahren verwenden zwei Schlüssel, einen privaten und einen öffentlichen. Nachrichten, die mit dem öffentlichen Schlüssel verschlüsselt wurden, können nur mit dem privaten Schlüssel entschlüsselt werden. Die Sicherheit der Public-Key Kryptographie beruht darauf, daß es nur mit großem Aufwand möglich ist, den einen Schlüssel aus dem anderen zu berechnen.

Digitale Unterschriften verwenden den umgekehrten Weg: Eine Nachricht wird mit dem privaten Schlüssel verschlüsselt; sie kann dann nur mit dem öffentlichen Schlüssel entschlüsselt werden. Ist die Nachricht bekannt, kann anhand des Ergebnisses überprüft werden, ob privater und öffentlicher Schlüssel zueinander passen. Als Nachricht wird ein eindeutiger Hashwert<sup>7</sup> (bijektive Abbildung) verwendet; nur identische Dokumente haben einen identischen Hashwert. Dazu werden alle für ein Java Applet notwendigen Klassen zu einem Java Archive (JAR) gebündelt. Von diesem Archiv wird mit einer Hashfunktion ein eindeutiger Hashwert berechnet, nur dieses Archiv hat diesen Hashwert. Änderungen an einer oder mehreren Klassen würden zu einem anderen Archiv mit einem anderen Hashwert führen, es kann also über den Hashwert überprüft werden, ob Änderungen durchgeführt wurden. Der Hashwert wird mit dem privaten Schlüssel verschlüsselt und im Header der Datei mit dem Archiv gespeichert.

Um das Applet zu autorisieren, berechnet die JVM des Benutzers zum einen den Hashcode des JAR-Files und entschlüsselt mit dem öffentlichen Schlüssel zum anderen den als digitale Unterschrift verwendeten Hashcode. Stimmen die beiden Werte überein, ist durch den öffentlichen/privaten Schlüssel sichergestellt, von wem das Java Applet kam und daß es nicht modifiziert wurde.

Der öffentliche Schlüssel kann entweder von einer Certificate Authority wie z.B. Verisign abgerufen werden, oder aber in einer lokalen Schlüsseldatenbank untergebracht sein. Ersteres Verfahren wird von Netscape und Mi-

---

<sup>7</sup>Eine Hashfunktion  $h$  bildet den Wert  $x$  mit einer bijektiven Abbildung auf einen Funktionswert  $h(x)$  ab. Aufgrund der Bijektivität gilt:  $h(x_1) \neq h(x_2) \forall x_1 \neq x_2$ .

crosoft implementiert, letzteres von Sun. Es wird das JVM Plugin von Sun verwendet, damit wird auch zwangsläufig das Sun Konzept mit der lokalen Schlüsseldatenbank verwendet.<sup>8</sup>

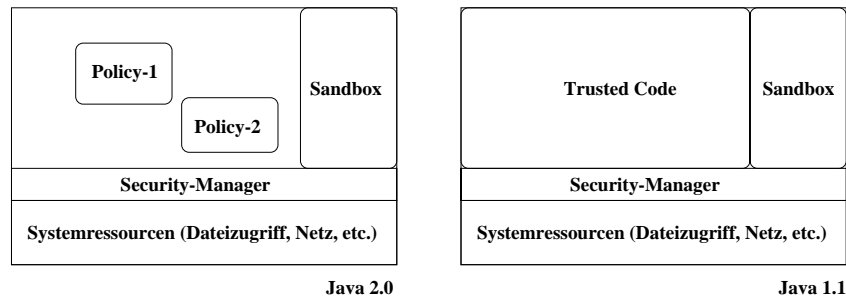


Abbildung 3.9: *Sicherheit unter Java 1.1 und Java 2.0: Unter Java 1.1 haben Klassen entweder alle Privilegien (trusted), oder keine (sandbox). Java 2.0 kann die Privilegien feinkörniger vergeben. Mit einer Policy können Privilegien für einzelne Klassen vergeben werden.*

Java 1.1 und Java 2.0 verwenden unterschiedliche Sicherheitskonzepte. Unter 1.1 kann der Benutzer einem Signed Applet entweder alle Privilegien einräumen, oder aber nur die Standardprivilegien, wie sie auch normale Applets besitzen (alles oder nichts). Es gibt keine Möglichkeit, die Rechte einzuschränken, um z.B. den Dateizugriff zu verbieten, das Öffnen von Netzwerkverbindungen zu einem bestimmten Rechner aber zu gestatten. Solche Erweiterungen sind unter Java 2.0 mit *Code Principals* möglich. Privilegien können sehr feinkörnig vergeben werden, es ist für jede Klasse einzeln festlegbar, welche Privilegien sie erhält (Datei, Netz, System.exec() Aufrufe) und es kann konfiguriert werden, zu welchem Rechner Netzwerkverbindungen aufgebaut und welche Dateien gelesen werden dürfen. Sicherheit unter Java und die unterschiedlichen Realisierungen sind ein komplexes Gebiet, das hier nur soweit vorgestellt wird, wie es bei der Implementierung von Bedeutung war. Die Details finden sich bei Sun, Netscape und Microsoft [Sun][DevEdge][Microsoft].

<sup>8</sup>Ab JVM Plugin 2.0 bietet Sun die Möglichkeit, den öffentlichen Schlüssel von einer Certificate Authority abzurufen. Die Funktion wird aber nur ab Service Pack 4 unter Windows unterstützt und wird deshalb nicht verwendet.

### 3.5.4 Java Applet Framework

Das Java-Applet bildet die Grundlage für die in den folgenden Abschnitten beschriebenen Anwendungen der statischen, dynamischen und quantitativen Telepathologie. Es wurde unter den Gesichtspunkten Anpassungsfähigkeit an den Benutzer und Erweiterbarkeit entworfen. Ziel der Anpassungsfähigkeit ist es, bezüglich der Anordnung der einzelnen Komponenten einer möglichst großen Anzahl von Benutzern gerecht zu werden. Jede grafische Komponente wird in einem eigenen MDI-Fenster (*internal frame*) dargestellt, das sich auf dem Desktop des Applets verschieben und verändern läßt (Abb. 3.10). Die Position der Fenster wird auf dem Server gespeichert und beim nächsten Login des Benutzers wiederhergestellt.

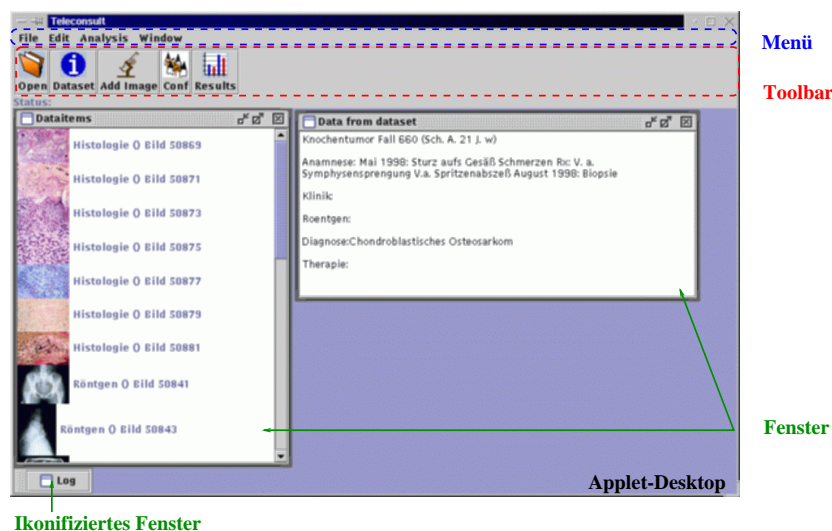


Abbildung 3.10: Die Benutzeroberfläche des Applets mit Menü, Toolbar und Applet-Desktop.

Die einfache Erweiterbarkeit des Applet Frameworks beruht auf der Trennung zwischen grafischen und funktionalen Klassen, die über einen Softwarebus<sup>9</sup> miteinander verbunden sind. Die Aufteilung ist in Abb. 3.11 gezeigt. Der ActionManager ist die zentrale funktionale, d.h. nicht-grafische Klasse.

<sup>9</sup>Ein Softwarebus ist ein Kommunikationsmechanismus zum Verteilen von Nachrichten. An den Softwarebus angemeldete Objekte werden informiert, sobald eine andere Klasse Daten über den Softwarebus verteilt.

Der ActionManager wurde als Singleton implementiert und ist für die Kommunikation mit dem Server und der Datenbank zuständig. Der Softwarebus verbindet den ActionManager mit den grafischen Komponenten. Grafische Komponenten können einen Menüeintrag, ein Icon in der Toolbar und ein Fenster haben. Alle Komponenten sind von der JFC Klasse ABSTRACT-ACTION abgeleitet. Diese stellt die Basisfunktionalität für einen Menü- oder Toolbareintrag bereit. Im Konstruktor wird eine Referenz auf den ActionManager und auf das Applet-Desktop (JAVAX.SWING.JDESKTOPPANEL) übergeben.

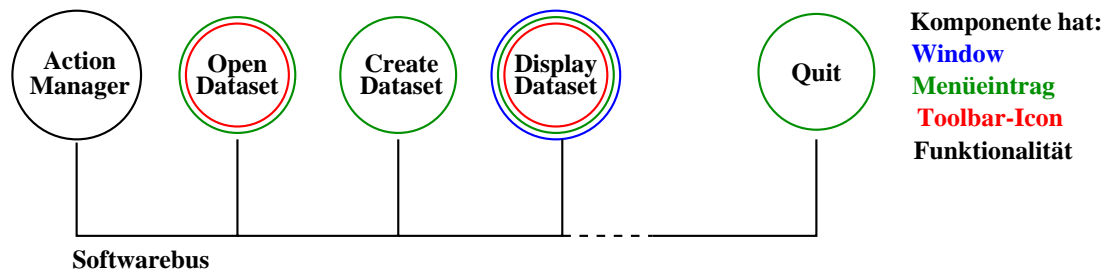


Abbildung 3.11: Die Komponenten des Applets sind aufgeteilt in Komponenten für die Funktionalität (ActionManager) und grafische Komponenten, die über einen Softwarebus miteinander verbunden sind.

Die Anpassung des Java Applets an unterschiedliche Anwendungen erfolgt über eine Konfigurationsdatei: Im Gegensatz zu C++ erlaubt Java eine dynamische Instanzierung von Klassen anhand ihres Klassennamens.<sup>10</sup> In der Konfigurationsdatei befinden sich die Klassennamen der zu instanzierenden grafischen Komponenten mit den entsprechenden Informationen über Menü- und Toolbar Eintrag.

### 3.6 Telepathologie: ein Überblick

Auf das Applet Framework aufbauend, werden in den folgenden Kapiteln Erweiterungen für die statische, dynamische und quantitative Telepathologie vorgestellt. Abbildung 3.12 zeigt eine Übersicht über die Architektur des

<sup>10</sup>Mit `java.lang.Class.forName(String className)` wird eine `java.lang.Class` Instanz erzeugt. Diese lässt sich mit der Methode `newInstance()` instanzieren.



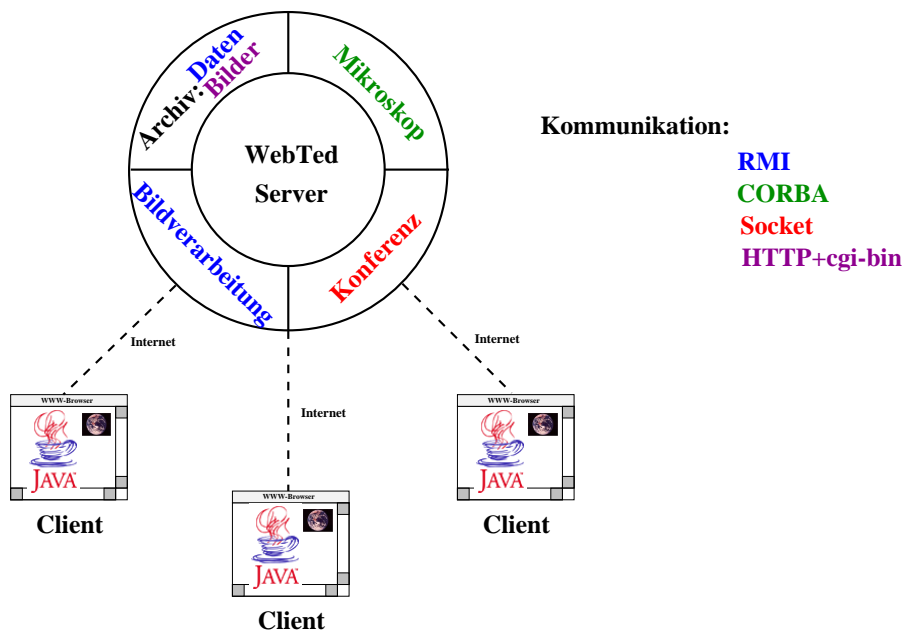


Abbildung 3.12: *Architektur des Systems. Auf dem Server stehen Dienste für Datenarchivierung, Bildverarbeitung, Konferenzen und Mikroskopansteuerung dem Client zur Verfügung.*

Servers. Auf dem Server stehen dem Applet Module für Datenbankdienste, Konferenzdienste, Bildverarbeitungsdienste und Dienste für die Ansteuerung des Mikroskops zur Verfügung. Die Kommunikation zwischen Client und Server erfolgt je nach Modul mittels RMI, CORBA, Socket oder HTTP. Bevor in den nächsten Kapiteln detailliert auf die einzelnen Anwendungen eingegangen wird, wird an dieser Stelle noch ein kurzer Überblick über die einzelnen Anwendungen und die jeweils verwendeten Module auf dem Server gegeben:

**Statische Telediagnostik:** Für die statische Telediagnostik werden klinische Daten zusammen mit den zugehörigen Bilddaten vom Java Applet aus auf dem Server abgelegt. Die Bilder können dabei entweder direkt über die Videokamera eines am Rechner angeschlossenen Mikroskops aufgenommen werden, oder aber aus einer Datei gelesen werden. Der Benutzer räumt dem Java Applet hierzu besondere Privilegien ein (Signed Applet, siehe S. 48).

Ein Konsultant kann sich dann den Fall ansehen und seine Diagnose ebenfalls über das Java Applet eingeben. Zusätzlich gibt es die Möglichkeit, den Fall in einem Konferenzmodus mit einem Kollegen zu besprechen. In diesem Modus sehen alle Teilnehmer an einer Konferenz das gleiche Bild und können während einer Diskussion mit Pfeilen auf interessante Merkmale im Bild hinweisen. Für die Sprachkommunikation wird eine direkte Telefonverbindung zwischen den Teilnehmern verwendet.

Für die Kommunikation mit dem Server wird für Textdaten (klinische Daten) Java RMI (blau) verwendet (Abb. 3.12). Die Bilder werden im Browser angezeigt, dazu wird auf dem Server mit einem cgi-bin Skript eine HTML-Seite generiert, die das Bild enthält. Die Bilddaten werden dann per HTTP-Protokoll (violett) an den Client verschickt. Die Daten einer Konferenzsitzung (Einladungen, Position der Pfeile) werden per Socketverbindung (rot) über das Konferenzmodul des Servers an die Clients verschickt. Eine direkte Kommunikation zwischen den Clients ist aufgrund der Sicherheitsmerkmale von Java nicht möglich, die Kommunikation zwischen den Clients läuft immer über das Konferenzmodul des Servers. Die statische Telediagnostik wird in Kapitel 4 vorgestellt.

**Dynamische Telediagnostik:** Bei der Arbeitsgemeinschaft Knochentumoren im Deutschen Krebsforschungszentrum steht ein motorisiertes Mikroskop (Leica DM IRBE). Mit dem Java Applet lassen sich alle wesentlichen Funktionen des Mikroskops (Vergrößerung, Ausschnitt, Fokus, Licht etc.) ferngesteuert bedienen. Für die Kommunikation zwischen Client und Server wird an dieser Stelle CORBA (grün) verwendet (Abb. 3.12). Es wäre auch eine Kommunikation über RMI möglich gewesen, aber aus technischen Gründen (Firewall) war an dieser Stelle eine CORBA Implementierung leichter zu realisieren gewesen. Details hierzu finden sich in Kapitel 5.

**Quantitative Telepathologie:** Auf dem Server abgelegte Bilder lassen sich durch verschiedene Bildverarbeitungsmethoden analysieren, um dia-

agnostische Parameter zu extrahieren. Die Ergebnisse werden vom Server abgerufen, mit einem cgi-bin Skript grafisch aufbereitet und im Browser dargestellt.

Neben diesem Anwendungsmodus gibt es noch den Entwicklungsmodus. Mit einer visuellen Entwicklungsumgebung lassen sich neue Verfahren entwickeln. Für die unterschiedlichen Schritte Filterung, Segmentierung und Merkmalsextraktion existieren speziell für die Pathologie entwickelte Module, die grafisch zusammengesetzt werden können. Fertig entwickelte Verfahren können in der Methodendatenbank abgelegt werden, um dann für die quantitative Auswertung eingesetzt zu werden.

Für die Kommunikation zwischen Client und Bildverarbeitungsserver wird Java RMI verwendet. Der Bildverarbeitungsteil wird in Kapitel 6 detailliert beschrieben.

## 3.7 Zusammenfassung

In Kapitel 3 wurden die Anforderungen an ein System für Telediagnostik und quantitative Telepathologie formuliert, und es wurde begründet, daß ein Client/Server Ansatz dafür eine sinnvolle Architektur darstellt. Aus praktischen Gesichtspunkten wurde ein webbasierter Ansatz gewählt. Es wurden die momentan verfügbaren Client/Server Technologien im Webbereich diskutiert und der Grobaufbau des WebTed-Systems begründet und vorgestellt.

In den folgenden Kapiteln werden Erweiterungen des in diesem Kapitel vorgestellten Applet Frameworks für die statische, dynamische und quantitative Telepathologie vorgenommen. Eine Übersicht über den Server findet sich in Anhang A.



# Kapitel 4

## Statische Telediagnostik (StaTed)

In diesem Kapitel wird die StaTed (Statische Telediagnostik)-Komponente vorgestellt. Die statische Telediagnostik wird für Telekonsultationen unter Experten eingesetzt. Um einen Fall konsultieren zu können, werden klinische Daten zusammen mit den zugehörigen Bilddaten vom Referenten auf dem Server gespeichert. Der Konsultant betrachtet den Fall mit dem in diesem Kapitel vorgestellten Java Applet und einem Webbrowser.

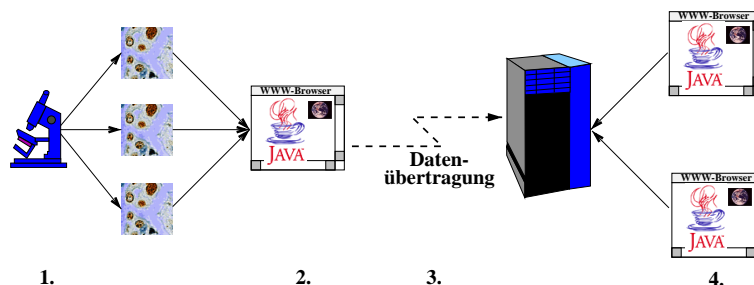


Abbildung 4.1: *Statische Telepathologie: Von einem Präparat werden repräsentative Ausschnitte ausgesucht (1), zu einem Fall zusammengestellt (2) und zum Server übertragen (3). Konsultanten werden per e-Mail benachrichtigt und betrachten den Fall mit ihrem Webbrowser (4).*

Am Anfang von Kapitel 4 wird ein Überblick über das Java Applet für die statische Telepathologie gegeben. Anschließend wird das der statischen Telepathologie zugrundeliegende Datenmodell erläutert und die Komponenten für die Bildaufnahme, Darstellung sowie die Konferenzkomponente vorgestellt.

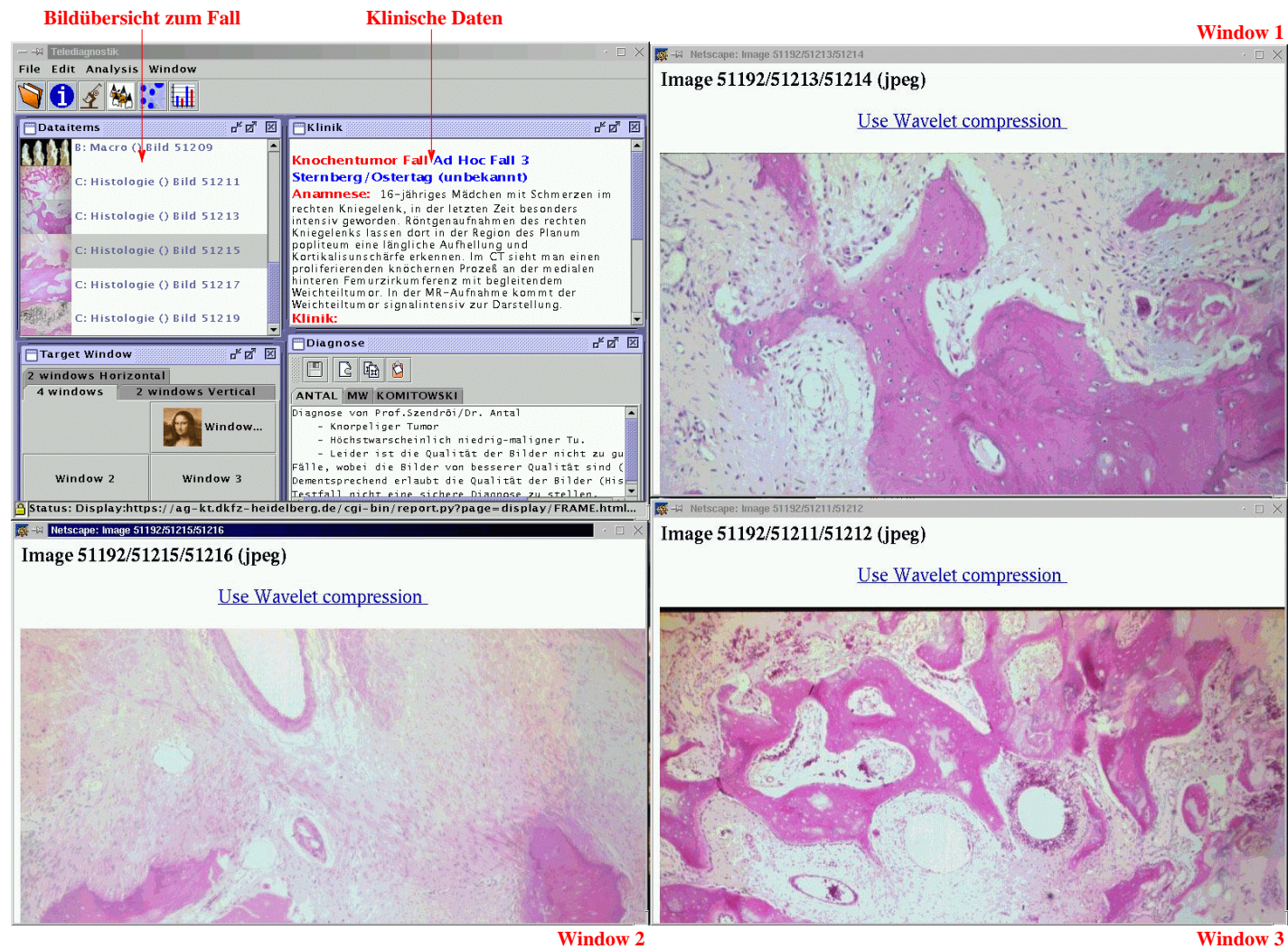


Abbildung 4.2: Benutzeroberfläche zur statischen Telepathologie. Außer dem Applet Fenster (oben links) gibt es noch 1–3 Browserfenster, um Bilder anzuzeigen.

## 4.1 Java Applet

Abbildung 4.2 zeigt die Benutzeroberfläche für die StaTed-Komponente. Die StaTed-Komponente erweitert das im vorigen Kapitel vorgestellte Applet Framework um Funktionalitäten für Bildaufnahme, Bilddarstellung und die Möglichkeit, Konferenzsitzungen abzuhalten. Daten, wie z.B. klinische Daten oder Bilddaten, werden in einzelnen, frei verschiebbaren Fenstern angezeigt.

### 4.1.1 Neue Fälle anlegen

Ein neuer Fall läßt sich mit dem Menüpunkt *File/New Dataset* anlegen. Es wird ein Formular geöffnet, in das sich die klinischen Daten eingeben lassen (Abb. 4.3b). Je nach ausgewähltem Datensatztyp unterscheiden sich die Felder eines Formulars. Die Daten zum neu angelegten Fall erscheinen in dem Fenster mit klinischen Daten.

Mit dem Menübefehl *Add Image* lassen sich Bilder aus unterschiedlichen Datenquellen (Datei, Framegrabber) zum angezeigten Fall hinzufügen. Dazu wird der in Abb. 4.3e dargestellte Dialog angezeigt. Bei der Auswahl von Dateien können entweder ein einzelnes Bild oder alle Bilder in einem Verzeichnis ausgewählt werden. Zu jedem Bild lassen sich weitere Daten (z.B. Färbung von histologischen Bildern) angeben (Abb. 4.3f). Mit den Menübefehlen *Edit Dataset/Edit Dataitem* werden die Daten vom Eigentümer nachträglich editiert. Es wird dasselbe Formular, wie beim Anlegen des Datensatzes, verwendet.

Der Besitzer des Datensatzes kann den Zugriff auf den Datensatz für einzelne Gruppen gewähren. Dies erfolgt über den Menübefehl *Permissions* (Abb. 4.3d). Gruppen lassen sich mit dem Menübefehl *Create Group* erzeugen und mit *Edit Group* editieren. Dazu wird ein Dialog mit allen Benutzern angezeigt, um eine neue Gruppe zu definieren (Abb. 4.3g). Der Benutzer wird Eigentümer dieser Gruppe und kann sie später verändern.

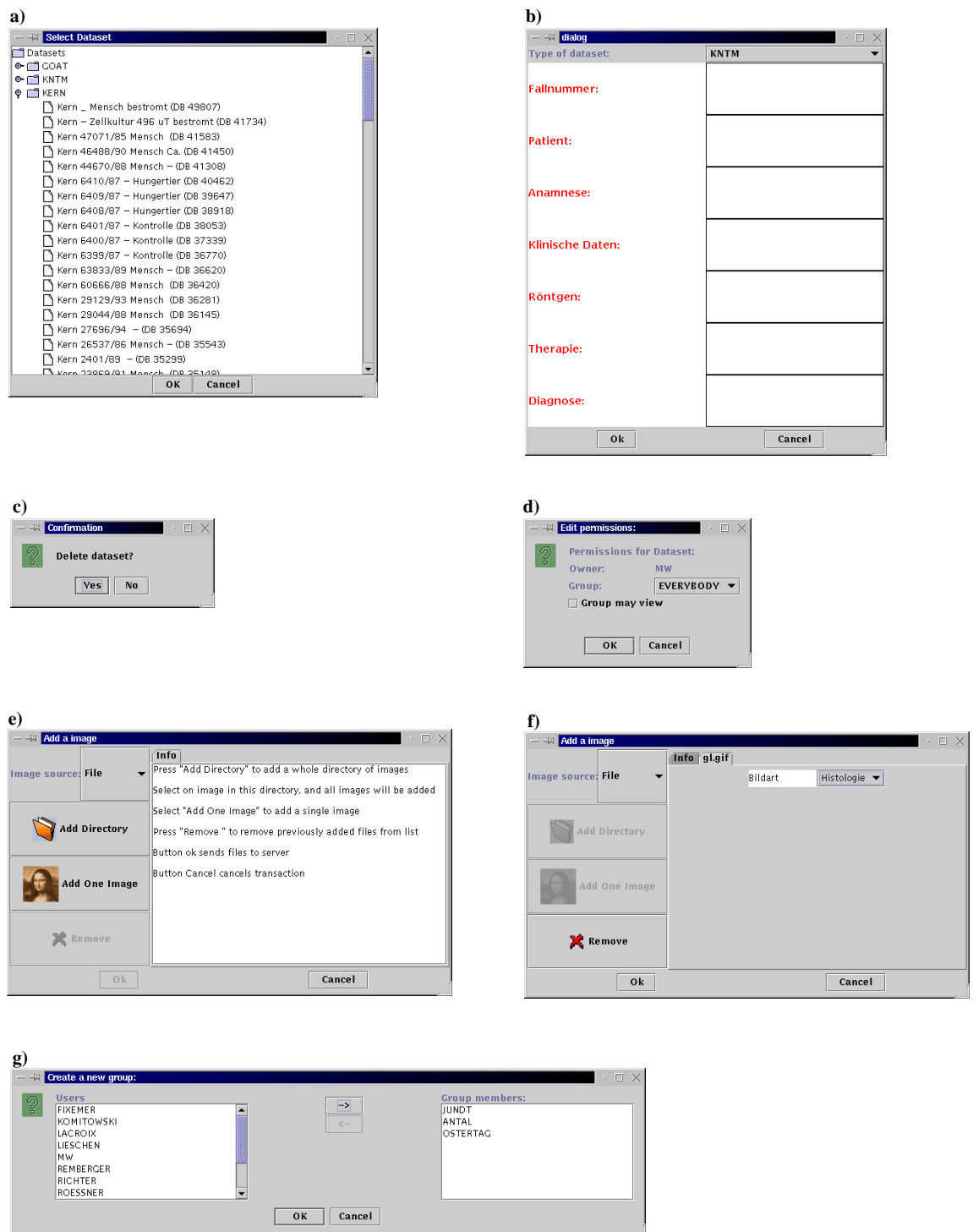


Abbildung 4.3: Widgets zur Benutzerführung.



### 4.1.2 Fälle abrufen

Mit dem Menübefehl *Open* wird ein Dialog (Abb. 4.3a) angezeigt, um einen Fall auszuwählen. Die unterschiedlichen Falltypen (GOAT – Gonarthrose, KNTM – Knochentumor, KERN – Kerngrading) sind in einen Baum einsortiert. Die angezeigten Namen werden mit einem Template generiert und erhalten die für den jeweiligen Typ relevanten Informationen, wie z.B. Name des Einsenders und Fallnummer (Abb. 4.3a).

Nach dem Öffnen eines Falles werden eine Liste mit verfügbaren Bildern sowie die klinischen Informationen zu dem Fall angezeigt. Durch einen Klick auf ein Bild wird das Bild in einem Fenster angezeigt.

Es sind verschiedene Anordnungen der Fenster möglich: Applet und drei Fenster zum Anzeigen der Bilder, Applet und ein Fenster für Bilder horizontal angeordnet, Applet und ein Fenster für Bilder vertikal angeordnet. Im Fenster *Target Window* wird angegeben, in welches Fenster das nächste Bild geöffnet wird. Somit ist es möglich, mehrere Bilder zum Vergleich in Fenstern nebeneinander anzuordnen.

Die Diagnose zu einem Fall läßt sich in einem speziellen Diagnose Fenster eingeben (Abb. 4.2). Jeder Benutzer kann seine Diagnose direkt in dieses Fenster eingeben bzw. zu einem späteren Zeitpunkt editieren. Die Diagnosen anderer Benutzer erscheinen unterhalb des Reiters mit dem Benutzernamen. Die Diagnosen sind für alle Benutzer, die den Fall sehen dürfen, lesbar.

Neben den Menüs gibt es noch Funktionen, die in der Toolbar untergebracht sind, wie z.B. das Whiteboard (Abschnitt 4.5).

## 4.2 Datenbankdienste

Eine der wichtigsten Anwendungskomponenten des Servers stellt die Datenbank dar. Sie enthält die klinischen Daten, Bilddaten, Ergebnisse quantitativer Auswertungen und Benutzerdaten. Als Datenbank wird die SQL-Datenbank Adabas 6 der Firma Software AG auf einem Linux-Server eingesetzt [Adabas].

Die Anbindung der Datenbank an den Client kann prinzipiell mit einer

2-Tier oder einer 3-Tier-Architektur erfolgen [Orfali 96]. Bei einer 2-Tier Lösung ist der Client direkt mit der Datenbank verbunden, bei einer 3-Tier Lösung kommuniziert der Client mit einer Zwischenschicht, der Database AbstractionLayer (DAL), die den Datenzugriff auf die SQL-Datenbank durchführt [Grotehen 95] (Abb. 4.4). Die vom Client verwendeten Datenstrukturi-

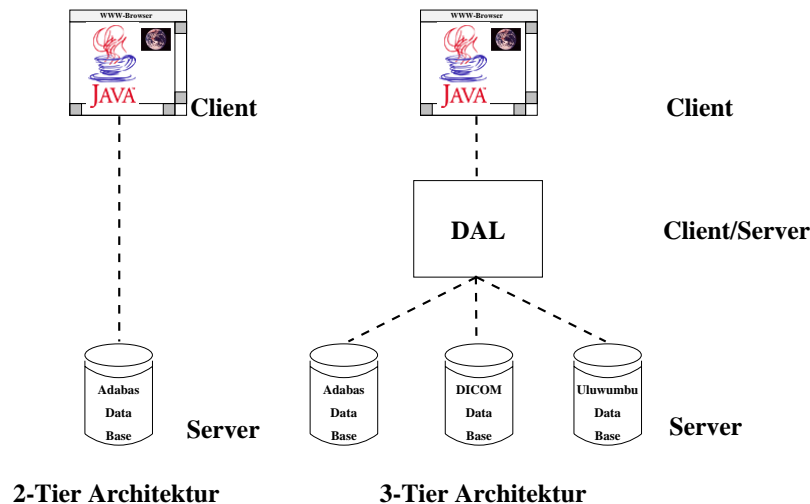


Abbildung 4.4: 2- und 3-Tier Architektur.

ren sind durch Objekte gekapselt, die per RMI mit der DAL kommunizieren, um Datensätze zu erzeugen, zu ändern, zu löschen etc.. Alle Datenbankoperationen werden von der DAL-Zwischenschicht ausgeführt; die Struktur der Datenbank bleibt für den Client hinter der Abstraktionsschicht verborgen. Für die Telepathologie bietet die Verwendung einer 3-Tier-Architektur mehr Möglichkeiten als eine 2-Tier-Architektur :

**Erweiterbarkeit:** Die DAL bietet eine einheitliche Schnittstelle zwischen Datenbank und Client. Die jetzige Implementierung mit einer Adabas SQL-Datenbank kann zu einem späteren Zeitpunkt z.B. gegen eine DICOM-Datenbank ausgetauscht werden, ohne daß der Client modifiziert werden muß.

**Kryptographie:** Bei einer 2-Tier-Architektur wird zwischen Client und Datenbank über eine TCP-Socketverbindung kommuniziert. Die meisten

Datenbanken, einschließlich der verwendeten Adabas Datenbank, sehen *keine* Möglichkeit eines verschlüsselten Kanals zwischen Datenbank und Client vor.

Eine 3-Tier-Architektur kann so implementiert werden, daß zwischen DAL und Client alle Daten verschlüsselt übertragen werden. Die Kommunikation zwischen DAL und Datenbank erfolgt lokal auf dem Server und benötigt daher keine Verschlüsselung.

Die Anforderungen des Bundesdatenschutzgesetzes lassen sich deshalb nur mit einer 3-Tier-Architektur erfüllen (s. Abschnitt 4.2.3).

**Caching:** Wiederkehrende Abfragen können von der Zwischenschicht gecached werden und erhöhen so die Performance.

**OO:** Die Tabellen der relationalen Datenbank können durch eine objektorientierte Darstellung gekapselt werden.

Durch den 3-Tier-Ansatz ergeben sich aber auch Nachteile: Der Client kann nicht mehr beliebige Anfragen ausführen, sondern ist auf die durch das Interface bereitgestellten Methoden beschränkt. Außerdem kann der ORB-Server bei vielen parallelen Anfragen zum Engpaß werden. Die Vorteile der 3-Tier-Architektur, insbesondere bezüglich Erweiterbarkeit und Kryptographie, überwiegen allerdings. Die Datenbankkomponente des WebTed-Systems wurde deshalb als 3-Tier Komponente realisiert.

### 4.2.1 Datenmodell

Aufgrund der fehlenden Standards in der Telepathologie wird ein proprietäres Datenmodell verwendet, das aber aufgrund der 3-Tier-Architektur sehr leicht gegen ein anderes Modell (z.B. DICOM) ausgetauscht werden kann. Hierzu muß lediglich serverseitig die Implementierung der DAL gegen eine DICOM o.ä. Implementierung ausgetauscht werden. Ein solcher Austausch wäre z.B. sinnvoll, sobald DICOM eine größere Verbreitung in der Pathologie gefunden oder sich ein anderer Standard etabliert hat.

In der Telepathologie wird ein *fallorientiertes* Datenmodell benötigt; es werden einzelne Fälle konsultiert, bei denen außer Alter und Geschlecht keine

Patientendaten benötigt werden, d.h. es werden anonymisierte Daten verwendet. Dies steht im Gegensatz zu *patientenorientierten* Datenmodellen, wie sie z.B. aus Patienteninformationssystemen in Kliniken bekannt sind, bei denen alle Informationen über einen Patienten zentral gespeichert werden, um den unterschiedlichen Fachabteilungen zur Verfügung stehen zu können.

Unterschiedliche Komponenten des WebTed-Systems haben unterschiedliche Anforderungen an das Datenmodell: Bei einem Fall, wie er typischerweise bei der AG-KT für Konsultationen anfällt, müssen klinische Daten wie z.B. Alter des Patienten, Anamnese, Klinik und Bilddaten (Makrobilder, Röntgenbilder und histologische Bilder) gespeichert werden. Zu jedem Fall soll jedes Mitglied der AG eine Diagnose eingeben können.

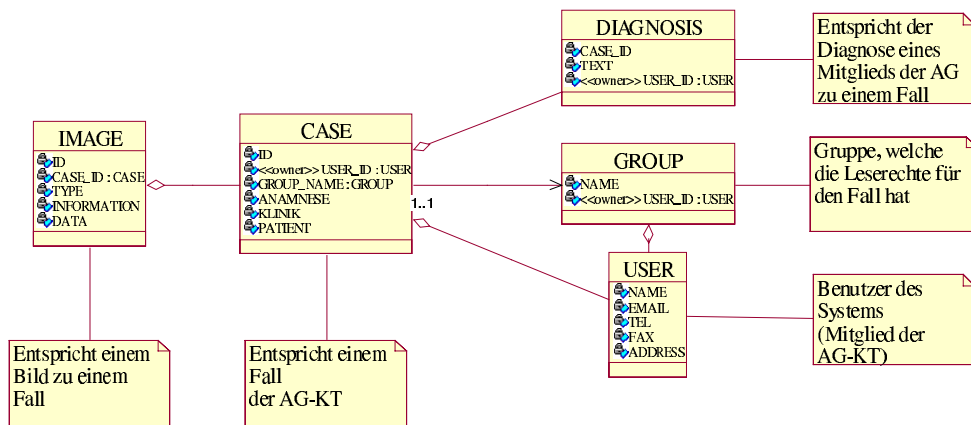


Abbildung 4.5: Ausschnitt eines Entity Relationship Diagramms (ER-Diagramm) für einen typischen Fall der AG-KT. Ein Fall (CASE) enthält klinische Daten (Patienteninformationen, Anamnese, Klinik etc.) und hat  $n$  Bilder (IMAGE). Zu jedem Fall kann jeder Nutzer des Systems eine Diagnose (DIAGNOSIS) erstellen.

Das in Abb. 4.5 gezeigte UML-Diagramm eines einfachen Datenmodells erfüllt diese Anforderungen. Das Modell ist leider nicht allgemeingültig; für andere Arten von Fällen sind andere Daten zu speichern. Beispielsweise werden für die quantitative Telepathologie zu jedem Bild numerische Daten gespeichert. Ein Teil der Anforderungen wird sich in Zukunft sicherlich ändern und verlangt ein flexibles Datenmodell mit *unterschiedlichen Typen* von Datensätzen, d.h. für die statische Telepathologie wird z.B. ein anderer

Datensatztyp verwendet, als für die quantitative Telepathologie. Eine Datenstruktur, die solchen Anforderungen gerecht wird, ist in Abb. 4.6 gezeigt. Sie besteht aus vier Grundentitäten, die Daten zu einem Fall und Daten zu den einzelnen Bildern enthalten.

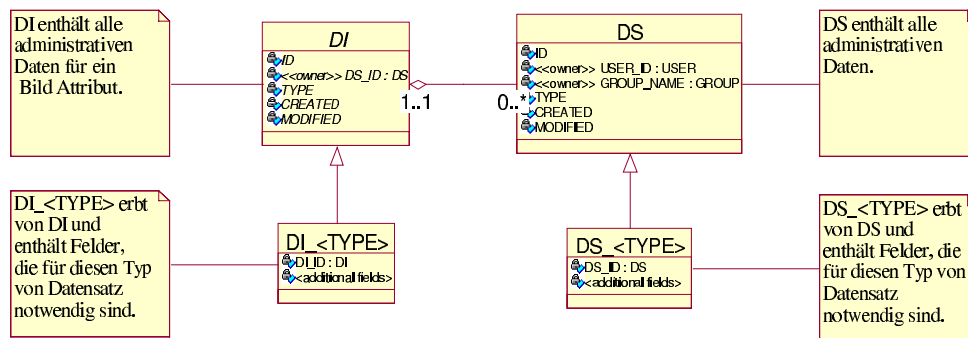


Abbildung 4.6: Ausschnitt eines ER-Diagramms um unterschiedliche Datensatztypen zu speichern. Ein konkreter Typ von Datensatz ( $DS\_<TYPE>$ ) erbt von einem Basisdatensatz (DS) und erweitert DS um weitere Felder. Zu jedem Datensatz gehören 1..n Dataitems (DI). Wie bei dem Dataset wird auch bei den Dataitems das Prinzip der Vererbung angewendet ( $DI\_<TYPE>$ ).

Den unterschiedlichen Anforderungen wird das Modell durch das Konzept der Vererbung gerecht. Dazu gibt es einen Typ DS, der alle administrativ notwendigen Daten wie z.B. Eigentümer, Typ, Gruppe, Rechte der Gruppe (lesen), Erzeugungsdatum und Änderungsdatum enthält. Um einen Datensatztyp zu realisieren, wird eine Unterklasse  $DS\_<TYPE>$  von DS gebildet, d.h. es wird aus diesen beiden Tabellen eine Sicht (SQL View) gebildet. Die Unterklasse enthält alle für diesen Datensatztyp notwendigen Felder; die administrativen Daten werden von DS vererbt. Zu jedem Dataset gibt es 0..n Dataitems (DI). Zu jeder Dataitem-Entität gehört eine für jeden Typ feste Anzahl von  $m$  Bildern, z.B.  $m=2$  für Bilddaten, die mit einem konfokalen Mikroskop mit 2 Kanälen aufgenommen wurden. Damit ergeben sich folgende Entitäten:

**Dataset (DS):** Der Dataset entspricht einem Fall. Es sind lediglich administrative Daten gespeichert (Eigentümer, Gruppe, Privilegien der Gruppe, Erzeugungsdatum).

**DS\_<TYP>:** Diese Entität ist von DS abgeleitet und enthält Daten, die für diese Art von Datensatz spezifisch sind. Beispielsweise enthält DS\_KNTM Felder für Patienteninformationen, Klinik, Anamnese etc.. Zwischen DS und DS\_KNTM besteht eine 1:1 Beziehung.

**Dataitem DI:** Ein Dataitem entspricht einem oder mehreren Bildern. DI und DS stehen in einer 1:n Beziehung. Ein Dataitem kann für 0..m Bilder stehen. In der Histologie können Schnitte mit verschiedenen Histomarkern angefärbt sein, die dann bei normalem und Fluoreszenzlicht zu unterschiedlichen Aufnahmen des gleichen Präparats führen. Sie werden im gleichen DI gespeichert.

**DI\_<TYP>:** Analog zum DS\_<TYP> können zusätzliche Daten zu den Bildern gespeichert werden.

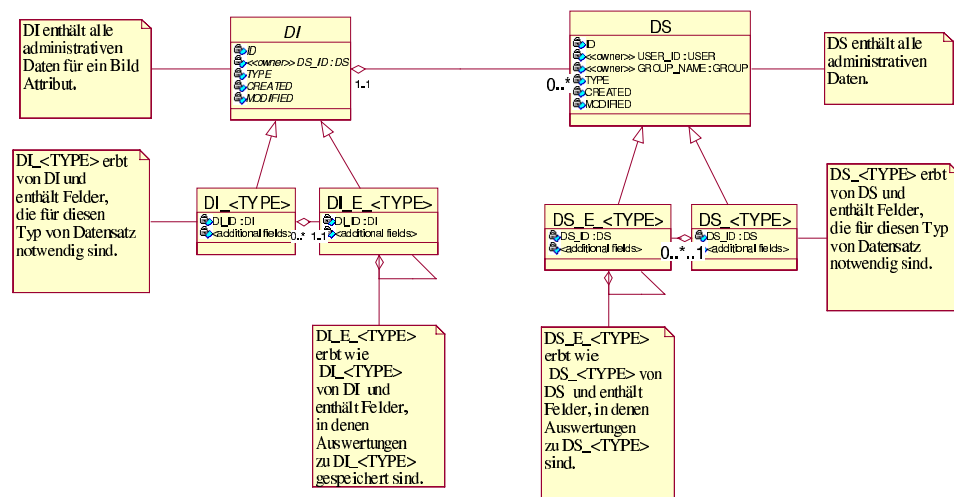


Abbildung 4.7: Ausschnitt des ER-Diagramms für die Implementierung von Knochentumordatensätzen (KNTM).

Eine Anwendung des Modells für Knochentumorfälle ist in Abb. 4.7 dargestellt. Für die Verarbeitung, z.B. Speicherung von Diagnosen zu Knochentumorfällen oder quantitativer Daten zu einem Bild, muß das Modell noch erweitert werden. Zu jedem Dataset oder Dataitem können noch Auswertungs-Datasets (Eval-Datasets) nach dem gleichen Vererbungsmuster hinzugefügt werden. Es gibt zwei Arten von Eval Datasets:

**DS\_E\_<TYP>:** Ist analog zu DS\_<TYP> von DS abgeleitet und enthält Daten, die spezifisch sind für die Auswertung von dieser Datensatzart. Beispielsweise enthält DS\_E\_KNTM ein Feld, um die Diagnose zu speichern. Zwischen DS\_<TYP> und DS\_E\_KNTM besteht eine 1:n Beziehung.

**DI\_E\_<TYP>:** Ist analog zu DS\_E\_<TYP>.

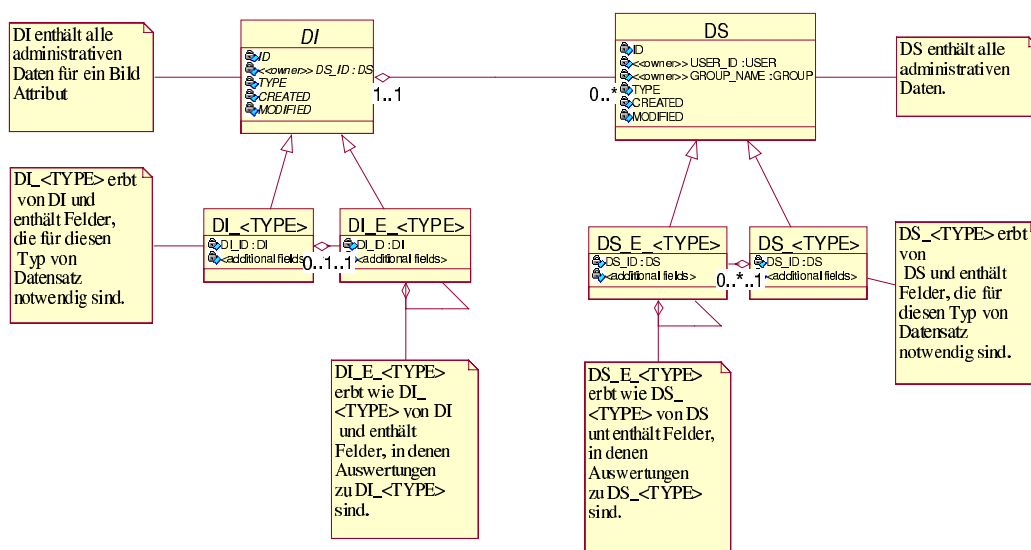


Abbildung 4.8: Ausschnitt des erweiterten ER-Diagramms. Zu jedem Datensatz bzw. Dataitem lassen sich noch Informationen (Daten zur quantitativen Auswertung, Diagnosen etc.) speichern (Kardinalität 0:n).

## 4.2.2 Zugriffsschutz

Der Zugriff auf Datensätze kann auf einzelne Benutzer oder Gruppen von Benutzern eingeschränkt werden. Eine Gruppe besteht aus 0..n Benutzern. In der Tabelle DS sind für jeden Datensatz Eigentümer, Gruppe und Rechte der Gruppe eingetragen. Der Eigentümer des Datensatzes kann der Gruppe die Leserechte auf den Datensatz geben. Damit haben alle Gruppenmitglieder Zugriff auf alle für den Datensatz gespeicherten Dataitems.

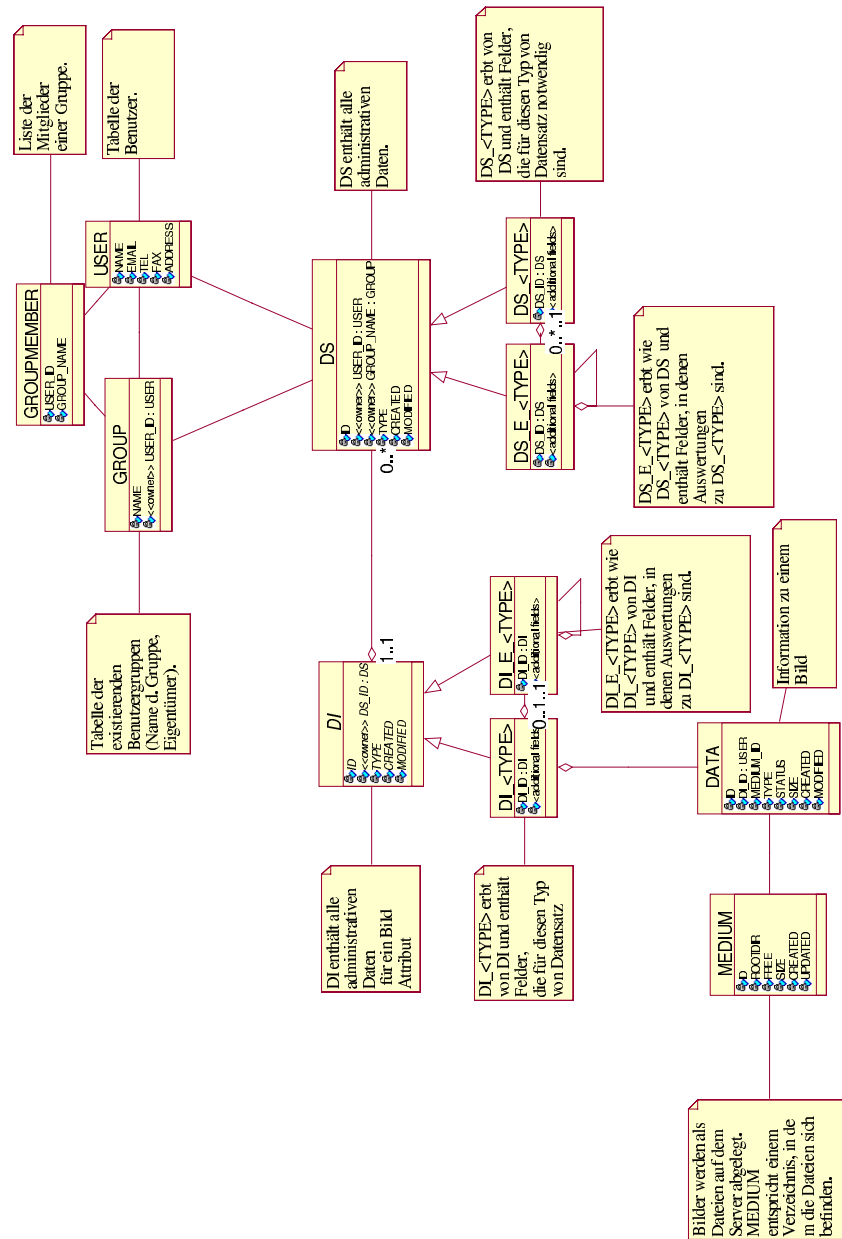


Abbildung 4.9: Komplettes Datenbankschema für die Telepathologie.



Für den Zugriffsschutz werden die in der SQL-Datenbank integrierten Sicherheitsaspekte verwendet. Eigentümer aller Tabellen ist der Datenbankadministrator (DBA). Der DBA besitzt Privilegien, um Tabelleneinträge einzufügen, zu löschen, zu ändern und um Privilegien an andere Benutzer zu vergeben. Das Adabas DBMS unterstützt die Vergabe von Privilegien nur auf der Ebene von Tabellen oder Sichten, d.h. die Einschränkung auf bestimmte Tabellenspalten ist nicht möglich.

Für jeden Benutzer wird deshalb von jeder Tabelle eine Sicht (View) erzeugt, in der nur die für den Benutzer freigegebenen Daten sichtbar sind. Der DBA gewährt nur diesem Benutzer den lesenden Zugriff auf die Tabelle. Zum Einfügen von Daten in eine Tabelle werden Stored Procedures verwendet. Stored Procedures sind Funktionen, die im Datenbank-Kernel ablaufen, um z.B. Tabelleneinträge zu modifizieren. Stored Procedures werden mit den Rechten des Eigentümers ausgeführt, es kann aber anderen Benutzern das Privileg zu ihrer Ausführung gegeben werden. Für jede Tabelle existieren deshalb Stored Procedures, um Tabelleneinträge einzufügen, ändern oder löschen zu können. Vor jeder Operation wird von der jeweiligen Stored Procedure überprüft, ob die Operation rechtmäßig ist.

### 4.2.3 Verschlüsselung

Wenngleich in der AG-KT Daten prinzipiell nur anonymisiert auf dem Server abgelegt werden, gibt es dennoch Gründe dafür, die Daten durch Verschlüsselung zu schützen: Zum einen lassen sich aus einer Kombination der Daten (Behandelnder Arzt, Alter) indirekte Rückschlüsse ziehen, zum anderen kann der Einsatz personenbezogener Daten in Zukunft nicht ausgeschlossen werden. Die speziellen Anforderungen gehen aus dem Bundesdatenschutzgesetz hervor. Auf eine genaue Analyse der Forderungen und ihre Umsetzung wird an dieser Stelle verzichtet, sie findet sich in Anhang C.

Das Bundesamt für Sicherheit in der Informationstechnik (BSI) veröffentlicht in regelmäßigen Abständen eine Abschätzung über die Sicherheit von gängigen kryptografischen Verfahren. Zur Zeit wird eine Schlüssellänge von mindestens 128 Bit, wie sie z.B. durch die Verfahren RC4 oder 3DES einge-

setzt werden, für die Zeit bis 2004 als sicher betrachtet [BSI]. Beide Verfahren werden vom SSL-Protokoll unterstützt, die Implementierung verwendet deshalb für die Kommunikation zwischen Client und Server SSL Socketverbindungen.

#### 4.2.4 Implementierung

Die DAL wird durch das Java Interface TELECONSULT.LANG.DBINTERFACE realisiert. Das Java Applet meldet sich beim WebTed-Server an und erhält vom RMI-ORB nach erfolgreicher Anmeldung mit Username/Password eine Referenz auf ein Objekt, welches das Interface DBINTERFACE aus dem Package TELECONSULT.LANG implementiert.

Für den Zugriff auf die Datenbank verwendet der Client die Klassen teleconsult.lang.Dataset und teleconsult.lang.Dataitem, teleconsult.lang.Image. Diese Klassen kommunizieren per RMI mit der DAL, welche die Schnittstelle zur Datenbank bildet.

### 4.3 Bildaufnahme

Zu einem Fall gehören neben den klinischen Daten selbstverständlich auch die Bilddaten unterschiedlicher Modalitäten. In erster Linie sind dies natürlich histologische Bilder, die von einer am Mikroskop angeschlossenen Videokamera aufgenommen werden. Aber auch Röntgenbilder, Makrobilder und Kernspinbilder sind für die Fälle der AG-KT notwendig.

Bilder werden mit dem Menübefehl *Add Image* zu einem Fall hinzugefügt. Als Bildquelle stehen entweder Datei oder Matrox Meteor Framegrabberkarte zur Verfügung (Abb. 4.10). Die Bildquellen können einfach erweitert werden, dazu muß lediglich ein Java Interface (teleconsult.lang.ImageSourceInterface) durch die Bildquelle implementiert und die Bildquelle in die Konfigurationsdatei aufgenommen werden.

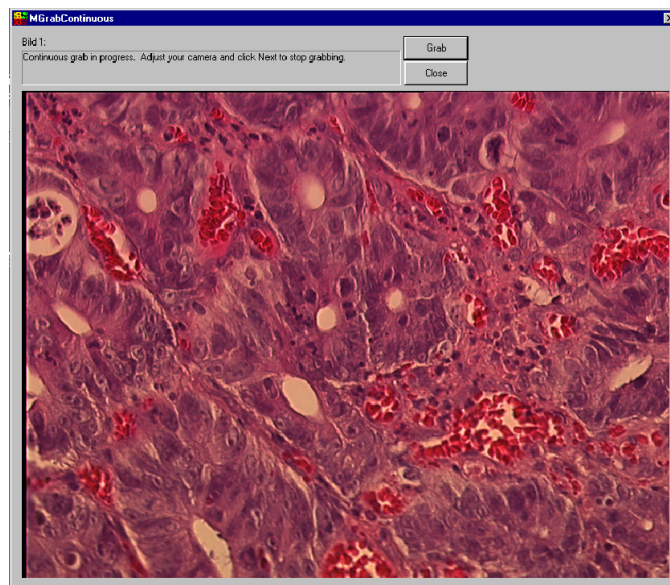


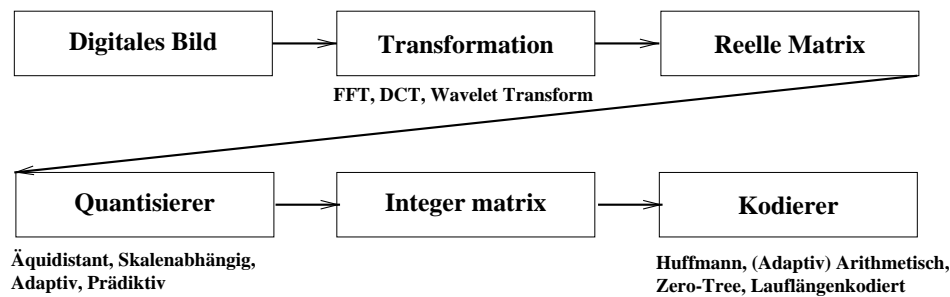
Abbildung 4.10: *Live-Videobild von der Matrox Meteor Framegrabberkarte.*

## 4.4 Bildkompression und Bilddarstellung

Eine schnelle Darstellung der vom Server abgerufenen Daten ist für die statische Telepathologie essentiell. Es ist dabei zwischen der Zeit, die für die Übertragung der Bilddaten vom Server erforderlich ist und der Zeit zur Dekompression und Darstellung der Bilder zu unterscheiden. Aufgrund der im allgemeinen niedrigen Bandbreite des Internets ist für eine effiziente Telepathologie die Kompression der Bilddaten erforderlich. Gängige Verfahren zur Kompression von Bilddaten sind JPEG- und Waveletkompressionen.

Die Kompression kann verlustlos oder verlustbehaftet sein; im ersteren Fall werden Kompressionsraten im Bereich 2–4 erreicht, im letzteren Fall bei immer noch akzeptabler Qualität können Kompressionsraten im Bereich 20–100 erreicht werden.

Das Schema für Bilddatenkompression ist in Abb. 4.11 dargestellt. Das digitale Bild wird mit einer Transformation in einen anderen Raum transformiert, in dem es eine kompaktere Darstellung hat, d.h. mit wenigen signifikanten Koeffizienten approximiert werden kann. Die JPEG-Kompression verwendet eine Cosinustransformation (DCT), die Waveletkompression eine

Abbildung 4.11: *Kompression von Bilddaten.*

Wavelettransformation. Der nächste Schritt ist die Quantisierung.

Die Transformation überführt die Integermatrix, die das Bild darstellt, in eine Matrix mit reellen Zahlen. Für eine effiziente Kodierung muß die Wertemenge der reellen Matrix reduziert werden. Dies kann entweder durch eine Einteilung in gleichgroße Bins (d.h.  $[0.5, 1.0) \rightarrow 1, [1.0, 1.5) \rightarrow 2$  etc.) oder adaptiv erfolgen. Die Quantisierung führt zu einer Reduktion der Informationsentropie und führt zu Verlusten in der Bildqualität.

Der letzte Schritt der Kompression ist die Kodierung der Daten. Es gilt dabei, Ähnlichkeiten, Wiederholungen und Muster in den Daten zu erkennen und sie dann mit einem kompakten Code darzustellen. Für die Waveletkompression sind solche Verfahren z.B. Embedded Zerotree, bei dem die Skalendarstellung der Wavelettransformation ausgenutzt wird, oder aber die arithmetische Kodierung, bei der Informationen über die statistische Verteilung von Koeffizienten ausgenutzt werden [Shapiro 93] [Buccigrossi 97] [Pearlman 96].

JPEG unterteilt ein Bild in Quadrate der Größe  $8 \times 8$  Pixel, die dann jeweils separat behandelt werden. Bei starken Kompressionen kommt es dann zu einer Blockstruktur.

Das Waveletkompressionsverfahren wendet die Waveletkompression global auf das ganze Bild an. Der Vorteil davon ist, daß es nicht zu Blockstrukturen kommen kann und daß eine progressive Übermittlung des Bildes möglich ist: Koeffizienten, die einen großen Beitrag liefern, werden zuerst geschickt, während kleine Koeffizienten (z.B. Rauschen) später geschickt werden. Eine progressive Übertragung erlaubt es dem Benutzer, die Qualität der Darstel-

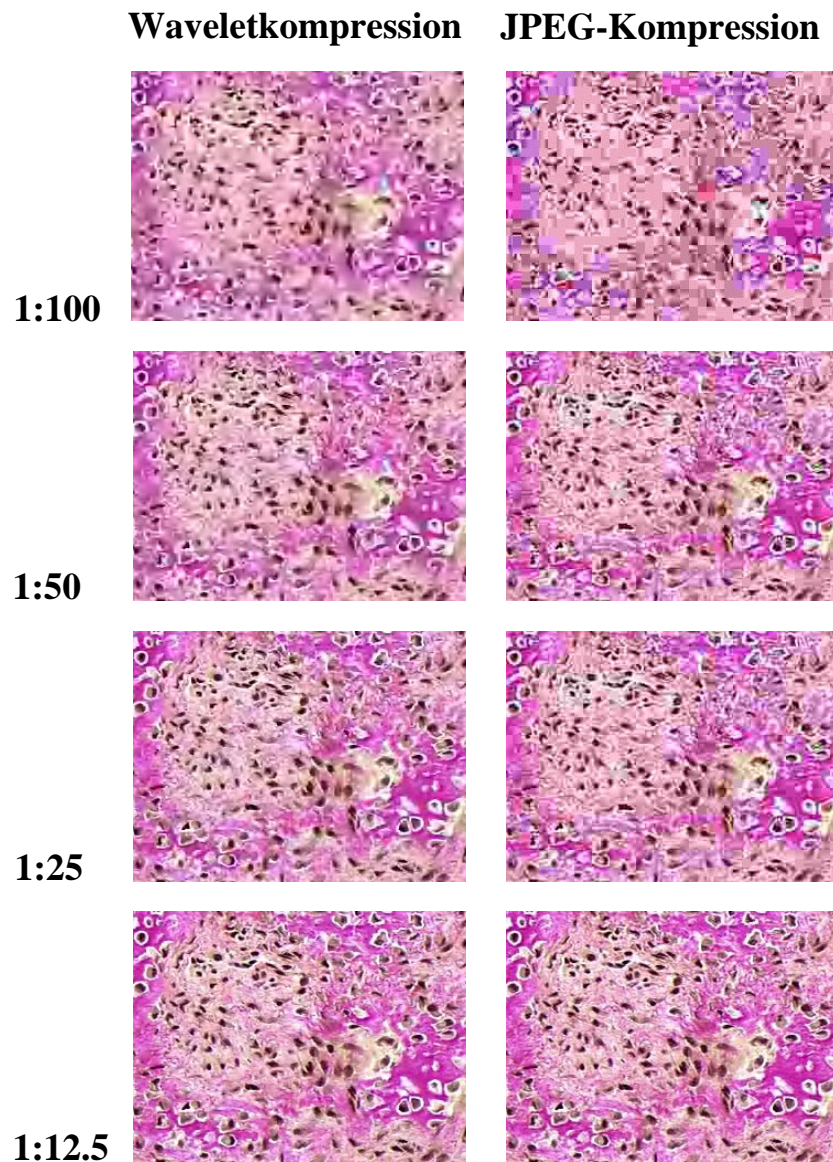


Abbildung 4.12: *Waveletkompression (links) und JPEG-Kompression (rechts) mit 100, 50, 25 und 12.5-facher Kompression.*



Abbildung 4.13: *Die JPEG Kompression zeigt bei hoher Kompressionsrate eine Blockstruktur (vergrößerte Darstellung).*

lung zu verbessern, indem noch weitere Koeffizienten vom Server geladen werden. Um gleiches mit der JPEG-Kompression zu erreichen, muß das Bild weniger stark komprimiert werden, und das Bild muß ein zweites Mal vom Server geladen werden.

In der Praxis erzielen Wavelet-basierte Kompressionsverfahren bei gleicher Größe eine etwas bessere Qualität als das JPEG-Kompressionsverfahren. Der Qualitätsgewinn wird aber mit einer erheblich langsameren Dekompression bei einem deutlich höheren Speicherbedarf erkaufte. Der erhöhte Speicherbedarf entsteht durch die globale Transformation, bei der das Bild mehrfach im Speicher gehalten wird. Auf einem langsamen Rechner mit einer schnellen Netzverbindung läßt sich ein JPEG-Bild sehr viel schneller darstellen, als ein Wavelet-Bild. Es besteht deshalb die Option, Bilder entweder als JPEG oder als waveletkomprimierte Bilder zu betrachten. Die JPEG-Kompression ist bereits im Browser integriert, für waveletkomprimierte Bilder wird das LuraWave-Plugin von LuraTech benötigt [Luratech].

## 4.5 Whiteboard

Telekonferenzen erfordern technische Hilfsmittel, um eine Diskussion möglichst effizient gestalten zu können. Bekannt dafür sind z.B. Videokonferen-

zen, in denen sich die Gesprächspartner gegenseitig sehen, oder Whiteboards, in denen ein Dokument bei allen Diskussionspartnern gleichzeitig angezeigt wird.

Typischerweise ist jeder Client Rechner durch eine Firewall gegen Zugriffe von außen abgeschirmt. Der Client darf Netzwerkverbindungen zu Rechnern außerhalb der Firewall aufbauen, der umgekehrte Weg ist allerdings ohne eine besondere Konfiguration der Firewall nicht möglich.

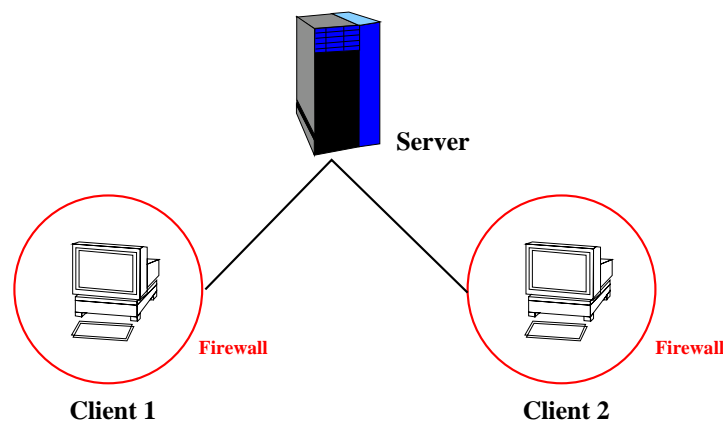


Abbildung 4.14: *Ein Client kann eine Verbindung mit einem anderen Client nur über ein Server aufnehmen. Die Firewall verhindert eine direkte Verbindung zwischen Client 1 und Client 2.*

In der Praxis müssen deshalb Daten für eine Konferenz immer den Umweg über einen Server machen: Client 1 und Client 2 bauen jeweils eine Verbindung zum Server auf. Der Server fungiert als Proxy und schickt alle Daten von Client 1 direkt an Client 2 weiter.

Für interaktive Konsultationen wurde ein Java basiertes Whiteboard implementiert. Videoconferencing wurde nicht implementiert, da es sinnvoll ist Videoconferencing auf das Java Media Framework (JMF) aufzubauen, das zum Zeitpunkt der Entwicklung noch nicht verfügbar war [JMF].

Die Implementierung des Whiteboards verwendet das Java Shared Data Toolkit (JSDT) [JSDT]. Es stellt die Basisfunktionalität für die Verteilung von Daten von einem Server an mehrere Clients zur Verfügung. Auf dem Server wird ein *Session*-Objekt instanziiert. Clients melden sich an das Session-Objekt an. Innerhalb jeder Session kann es eine beliebige Anzahl von



*Channels* geben, über die dann Daten verschickt werden. Channels und Sessions werden von Managern überwacht, die kontrollieren, welche Clients ein- und ausgeladen werden. Für die Synchronisation stehen *Tokens* zur Verfügung, die z.B. den Zugriff auf eine Ressource regulieren. Das JSdT kann unterschiedliche Netzwerkprotokolle verwenden. Zur Auswahl stehen das JSdT-Socket-Protokoll (TCP oder UDP), das Multicast-LRMP-Protokoll, RMI und HTTP zur Verfügung. Das Whiteboard der StaTed-Komponente verwendet die TCP Socket Implementierung.

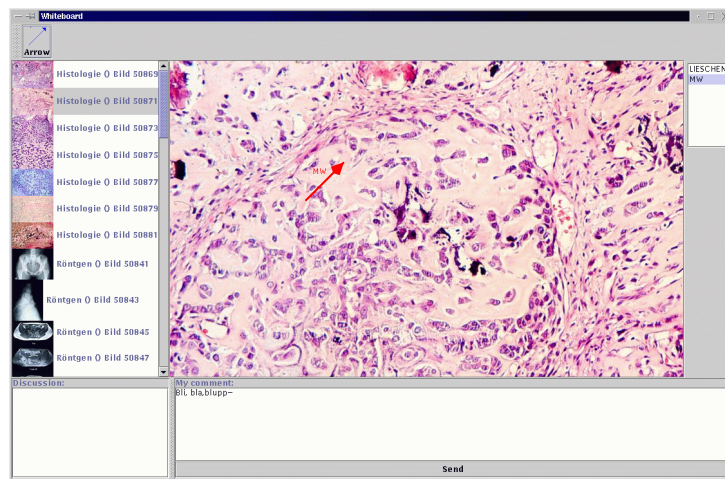


Abbildung 4.15: *Das Whiteboard der statischen Telepathologie.*

Für eine Konferenz ist eine vorherige zeitliche Absprache der Teilnehmer erforderlich. Zu Beginn einer Konferenz wird als erstes ein neues Session-Objekt auf dem Server kreiert. Der Initiator der Konferenz kann die einzuladenden Benutzer in einem Dialog auswählen. Jedem eingeladenen Benutzer wird ein Dialog mit der Frage, ob er an der Sitzung teilnehmen möchte, angezeigt. Nach der Anmeldung hat jeder Benutzer ein Fenster mit dem in Abb. 4.15 gezeigten Whiteboard. Der Initiator des Falles leitet die Diskussion. Er kann die Bilder des Falles auswählen, sie werden dann gleichzeitig bei allen Teilnehmern angezeigt. Alle Teilnehmer können durch Pfeile auf interessante Stellen im Bild hinweisen. Die Pfeile werden auf allen Whiteboards angezeigt.



### 4.5.1 Whiteboard Implementierung

Um die während einer Konferenzsitzung übertragene Datenmenge so gering wie möglich zu halten, ruft jeder Client die Bilddaten direkt vom Server ab. Lediglich Informationen über die im Bild eingeblendeten Pfeile werden über den Server an die Whiteboards der anderen Teilnehmer geschickt. Damit ergibt sich eine Datenmenge von etwa 500 Bytes pro angezeigtem Pfeil. Ein Channel des JSDT unterstützt nur den Versand der einfachsten Informationseinheit, eines Bytearrays. Dies ist aber vollkommen ausreichend, da die Serialisierung von Objekten in Java integriert ist. Wenn ein Benutzer Pfeile hinzufügt oder verändert, wird das Array mit Pfeilen serialisiert und als Bytearray über den Channel geschickt. Jedes Whiteboard empfängt das Array, stellt das Objekt aus dem Bytearray wieder her und überlagert das Bild mit den Pfeilen (Abb. 4.16).

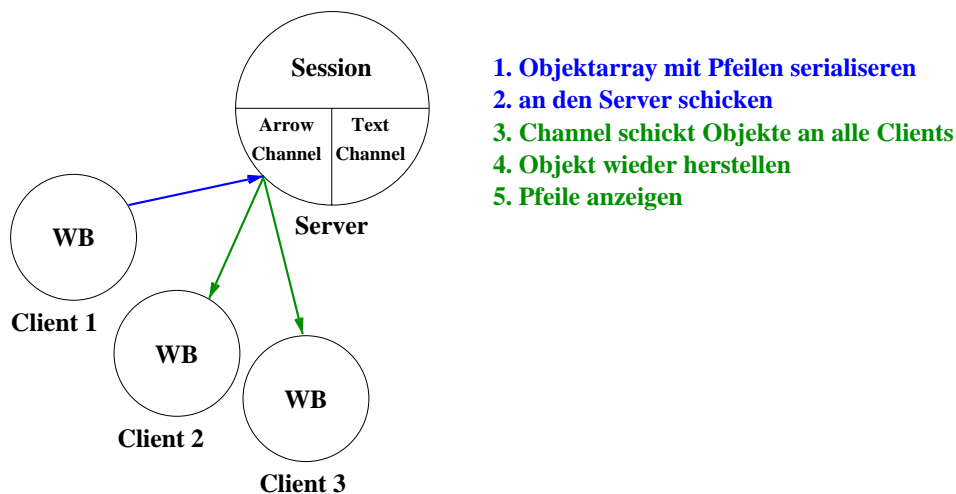


Abbildung 4.16: Datenverteilung zwischen unterschiedlichen Whiteboards (WB) mit dem JSDT: Nach Änderung einer Pfeilposition wird der Vektor mit den Pfeilobjekten serialisiert und über den Arrow Channel des Session Objekts an den Server geschickt. Der Server verteilt es über den Arrow Channel an alle angemeldeten Clients. Die Clients stellen den Vektor wieder her und zeigen die Pfeile im Whiteboard an.

### 4.5.2 Sprachübertragung

Prinzipiell wäre es möglich, auch Sprache über das Internet zu übermitteln. Ein Versuch mit der VocalTec Internet Phone 5 Software führte aber zu dem Schluß, daß die Technik dazu noch nicht ausgereift ist. Das Hauptproblem stellte die Sprachqualität dar; sie erreichte im lokalen Netzwerk allenfalls die Qualität eines Mittelwellensenders. Es wird deshalb davon ausgegangen, daß die Sprache per Telephon übertragen wird.

## 4.6 Zusammenfassung

Aufbauend auf das Applet-Framework wurde ein webbasiertes System für die statische Telepathologie vorgestellt. Mit einem Webbrowser lassen sich von einem Referenten neue Fälle anlegen und verwalten, Konsultanten werden per e-Mail über neue Fälle benachrichtigt und können ihre Diagnose direkt über das Web eingeben. Alternativ gibt es auch die Möglichkeit, Fälle in einer Konferenz zu besprechen. Dazu wird bei jedem Teilnehmer einer Konferenz ein Fenster angezeigt, in dem ein Bild dargestellt wird. Während der Diskussion kann mit Pfeilen, die bei allen Mitgliedern angezeigt werden, auf Besonderheiten im Bild hingewiesen werden.

# Kapitel 5

## Dynamische Telediagnostik (DynTed)

In Fällen, in denen kein Pathologe vor Ort verfügbar ist, um für eine Expertenkonsultation repräsentative Ausschnitte eines Falles zusammenzustellen, bietet sich die Nutzung von dynamischer Telediagnostik an. Das WebTed-System implementiert die dynamische Telediagnostik durch die DynTed (Dynamische Telediagnostik)-Komponente. Die dynamische Telediagnostik gestattet dem Konsultanten im Gegensatz zur statischen Telediagnostik die freie Auswahl des Bildmaterials mithilfe eines motorisierten Mikroskops, bei dem alle wesentlichen Funktionen ferngesteuert bedient werden können. Das Videobild einer am Mikroskop angeschlossenen Kamera wird über ein Netzwerk auf den Monitor des Konsultanten übertragen.

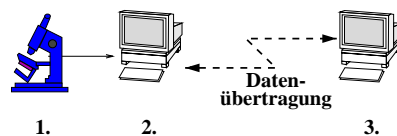


Abbildung 5.1: *Dynamische Telepathologie.*

Der Aufbau eines solchen Systems ist in Abb. 5.1 gezeigt. An einem motorisierten Mikroskop ist ein Steuerungsrechner angeschlossen, über den die unterschiedlichen Funktionen des Mikroskops angesteuert werden (Position

des x-y Tisches, Fokus, Licht). Zur Bildaufnahme wird eine am Mikroskop angeschlossene Videokamera in Kombination mit einer Framegrabberkarte im Steuerungsrechner verwendet. Der Steuerungsrechner fungiert als Server; eine Client-Software stellt die Bilder auf einem Monitor dar und erlaubt dem Konsultanten die Steuerung des Mikroskops. Auf die Vor- und Nachteile der statischen und dynamischen Telediagnostik bezüglich der Diagnosequalität soll an dieser Stelle nicht mehr eingegangen werden, sie wurden in Abschnitt 2.3.2 (S. 12) diskutiert.

## 5.1 Strategie zur Bildaufnahme

Den großen Nachteil der dynamischen Telediagnostik stellt die umfangreiche Datenmenge für die Videoübertragung dar; eine handelsübliche Videokamera arbeitet mit einer Auflösung von  $768 \times 576$  Pixel bei 24 Bit Farbtiefe. Dies ergibt bei einer Bildwiederholrate von 20 Hz einen Datenstrom von 25 MB/s. Selbst bei einer für Diagnosezwecke viel zu starken Kompression von 1:100 ergibt sich damit noch ein Datenstrom von 250 KB/s. Für die Verwendung von ISDN Leitungen werden deshalb Größe und Wiederholungsrate der Videobilder erheblich reduziert (z.B.  $352 \times 288$ , 7 Hz; vgl. [Kayser 99]).

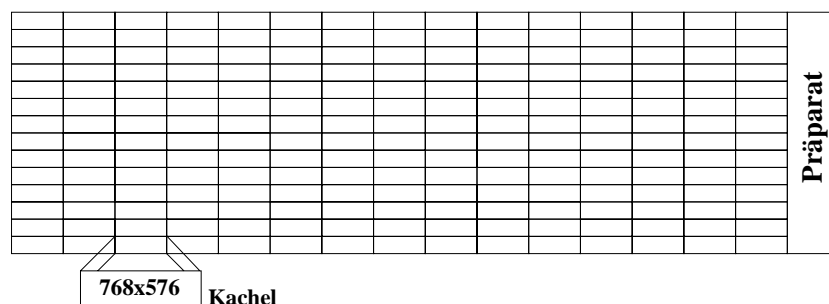


Abbildung 5.2: Das Präparat wird in Kacheln der Größe eines Videobildes ( $768 \times 576$  Pixel) aufgeteilt.

Im folgenden wird eine Strategie zur Bildaufnahme vorgestellt, die die Vorteile der statischen Telediagnostik (große Bilder) mit den Vorzügen der dynamischen Telediagnostik (freie Selektion der Ausschnitte) kombiniert: Anstatt von jeder neuen Position das Videobild zu übertragen, wird das

Präparat – wie in Abb. 5.2 gezeigt – in Kacheln aufgeteilt. Eine Kachel entspricht dabei der Größe eines Einzelbildes der Videokamera, d.h.  $768 \times 576$  Pixel. Die Kacheln werden jeweils nur einmal aufgenommen und im Client zu einem Bild zusammengesetzt. Wird der Bildausschnitt verschoben, werden die Kacheln im Client verschoben zusammengesetzt, und es muß nicht ein komplett neues Bild übertragen werden.

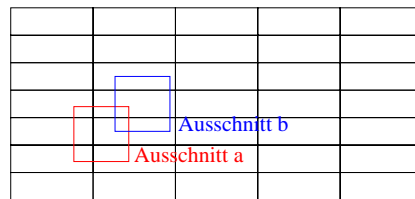


Abbildung 5.3: Wird im Client der Bildausschnitt verschoben, kann ein Teil der Kacheln wiederverwendet werden.

Der Client lädt jeweils ganze Bild-Kacheln vom Server und kombiniert sie transparent für den Benutzer zu einem passenden Bildausschnitt. Wird der Bildausschnitt verschoben (z.B. von Ausschnitt *a* zu Ausschnitt *b* in Abb. 5.3), können bereits vom Client angeforderte Kacheln wiederverwendet werden. Neue Kacheln werden vom Server angefordert, der dann das Mikroskop in die benötigte Position (x,y,z- Position, Licht, Objektiv) bringt und das Bild aufnimmt. Der Vorteil des Verfahrens besteht darin, daß ein beliebig großer Bildausschnitt aus Einzelbildern zusammengestellt werden kann, d.h. es sind auch erheblich größere Bilder möglich, die dann z.B. aus vier Einzelbildern zusammengestellt werden.

Grimes et al hat ein ähnliches Verfahren entwickelt, in dem jeweils Einzelbilder aufgenommen und angezeigt werden [Grimes 97]. Im Gegensatz zu dem hier vorgestellten Verfahren wird dabei das Bild im Client aber *nicht* aus unterschiedlichen Kacheln zusammengesetzt und hat deshalb die feste Größe  $768 \times 576$  Pixel.

### 5.1.1 Vor- und Nachteile

Die beiden dargestellten Verfahren zur Bildaufnahme haben ihre Stärken in unterschiedlichen Bereichen. Für eine Befundung wird ein Präparat bei un-

terschiedlichen Vergrößerungen betrachtet. Bei starken Vergrößerungen (Objektive  $25\times$ ,  $40\times$ ) muß aufgrund der geringen Schärfentiefe oft nachfokussiert werden. In solchen Fällen ist eine Übertragung als MPEG-Videosequenz sinnvoll, da aufgrund der hohen Redundanz von Bildern unterschiedlicher z-Position eine gute Kompression zu erwarten ist.

Prinzipiell wäre es auch möglich, das Präparat als Kacheln zu übertragen, allerdings müssen dann für jedes Bild typischerweise mehrere Kacheln aufgenommen werden. Die Motoren des Mikroskops sind in erster Linie auf Präzision und nicht auf Geschwindigkeit optimiert. Die Fahrzeit von einer Kachel zu nächsten beträgt je nach Vergrößerung zwischen 0.3 und 0.5 Sekunden und läßt deshalb keine hohe Interaktivität zu. Für die Übertragung aller für ein Bild notwendigen Ausschnitte sind daher mehrere Sekunden erforderlich. In solchen Situationen ist eine Videoübertragung sinnvoller.

Der umgekehrte Fall tritt bei geringen Vergrößerungen auf: Aufgrund der großen Schärfentiefe kommt es selten zu Änderungen des Fokus und vielmehr zu Änderungen der Position. Es bietet sich daher der semi-dynamische Ansatz mit Kacheln an. Zusätzlich können dabei im Hintergrund Kacheln von einer Umgebung des aktuellen Bildausschnittes gecached werden. Wird dann in dieser Umgebung ein neuer Bildausschnitt benötigt, kann auf die Kacheln schneller zugegriffen werden.

Beide Verfahren haben Vor- und Nachteile. Um die jeweiligen Stärken am besten nutzen zu können, besteht eine sinnvolle Strategie darin, bei geringen Vergrößerungen (Objektive  $8\times$ ,  $16\times$ ) das semi-dynamische Aufnahmeverfahren zu verwenden, und bei starker Vergrößerung ein MPEG-kodiertes Videobild zu übertragen.

## 5.2 Anforderungen

Bei der AG-KT wird die dynamische Telediagnostik für Expertenkonsultationen eingesetzt, d.h. sie soll die Diskussion zwischen Experten über einen Fall erleichtern. Die bestmögliche Qualität für die Beobachtung eines Präparats wird bei der Betrachtung unter dem Mikroskop erreicht; die Aufnahme mit

einer Videokamera, die anschließende Digitalisierung und Kompression bedeuten eine Verschlechterung der Bildqualität. Um die Diskussion zwischen Experten zu erleichtern, wird die dynamische Telediagnostik deshalb als Diskussionsmikroskop<sup>1</sup> eingesetzt, bei dem ein Experte direkt am Mikroskop arbeitet und ein anderer über einen Computer damit verbunden ist. Dabei sehen beide das gleiche Bild und haben beide die Möglichkeit, Einstellungen am Mikroskop zu verändern.

## 5.3 Implementierung

Für die Implementierung der dynamischen Variante stand das LEICA DM IRBE Mikroskop der AG-KT am DKFZ zur Verfügung, bei dem sich folgende Funktionen motorisiert bedienen lassen:

- x,y-Position des Objektives.
- Fokus (z-Position).
- Wahl des Objektivs im Objektivrevolver ( $2.5\times$ ,  $16\times$ ,  $25\times$ ,  $40\times$ ).
- Intensität der Beleuchtung.

Das Mikroskop ist zur Kommunikation mit einem Steuerrechner mit zwei seriellen RS232 Schnittstellen ausgestattet und erlaubt die Steuerung der oben genannten Funktionen. Es wird ein x,y Positionierungstisch der Firma Merzhäuser Wetzlar verwendet. Positionen können entweder computergesteuert oder aber manuell mit einem Joystick angesteuert werden. Als Genauigkeit für die Positionierung des x,y-Tisches wird  $0.1\mu m$  angegeben, der z-Trieb arbeitet mit einer Genauigkeit von  $0.05\mu m$ . Für die Bildaufnahme wird eine Sony DXC-950P CCD Farbkamera ( $768\times 576$  Pixel, 24 Bit Farbtiefe) und eine Matrox Meteor Framegrabberkarte eingesetzt.

---

<sup>1</sup>Als Diskussionsmikroskop wird ein Mikroskop mit mehreren Okularen bezeichnet. Jedes dieser Okulare ermöglicht den Blick auf das Präparat.

### 5.3.1 Vorarbeiten

Unter Anleitung des Autors hat Jochen Schröder in seiner Diplomarbeit ein System zur automatischen, zeitgesteuerten Aufnahme mit diesem Mikroskop entwickelt [Schröder 99]. Das System gestattet die Ansteuerung des x,y-Tisches, einen Autofokus und die Helligkeitsnormalisierung mit dem Java Programm CellRec. Allerdings ist es nicht als Client/Server-System für den Einsatz in der dynamischen Telediagnostik entwickelt worden.



Abbildung 5.4: *Photo des Leica DM IRBE Mikroskop der AG Knochentumoren.*

Die Philosophie des WebTed-Systems beruht auf einer komplett webbaasierten Telediagnostik, deshalb ist auch die DynTed-Komponente in Java geschrieben. Bei der Implementierung konnte die C-Bibliothek zur Ansteuerung des Mikroskops mit kleinen Modifikationen übernommen werden.

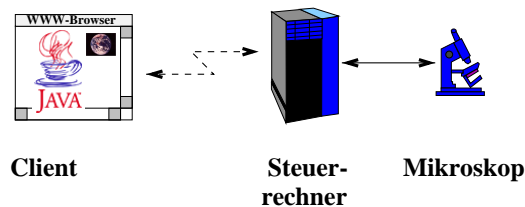


Abbildung 5.5: *Architektur der dynamischen Telepathologiekomponente: Vom Steuerrechner wird ein Java Applet zur Fernsteuerung des Mikroskops heruntergeladen.*



Die Architektur der DynTed-Komponente ist in Abb. 5.5 dargestellt. Für den Client wird, wie schon bei der statischen Telediagnostik, ein Java Applet als Benutzeroberfläche eingesetzt. Der Server stellt dem Java Client die Funktionalitäten zur Ansteuerung des Mikroskops, einen Autofokus und die Bildaufnahme in Form eines CORBA-Objekts zur Verfügung. Der Client erhält vom Webserver eine Referenz auf das CORBA-Objekt und kann damit die unterschiedlichen Funktionen des Mikroskops ansteuern.

### 5.3.2 Firewall

An dieser Stelle wäre es aus technischer Sicht naheliegend gewesen, das schon bei der statischen Telediagnostik eingesetzte Java RMI zu verwenden: Aus der Arbeit von J. Schröder existierte bereits eine Java-Bibliothek, die mit dem JNI (Java Native Interface) mit einer C-Bibliothek das Mikroskop ansteuerte. Der Client ist ebenfalls in Java geschrieben, so daß dies ideale Rahmenbedingungen für Java RMI wären. Das Problem stellt an dieser Stelle die Firewall des DKFZ dar: Sie stellt keine Restriktionen für den Netzwerkverkehr von Rechnern innerhalb des DKFZ nach außen dar, läßt aber nur in begründeten Fällen Verbindungen von außen nach innen zu. Für diese begründeten Fälle werden Regeln der Form *Verbindungen von IP-Adresse nach IP-Adresse auf Port x* aufgestellt.

Das Problem bei der Verwendung von Java RMI ist nun folgendes: Ein Serverobjekt meldet sich unter einem bestimmten Namen bei der Java Registry an. Ein Client-Objekt fragt bei der Registry nach einem Objekt mit diesem Namen und erhält, wenn vorhanden, einen Objekt-Stub. Der Objekt-Stub hat die Funktion eines Proxys und kommuniziert mit dem Objekt-Skeleton, der in Funktion eines Proxys beim jeweiligen Objekt die Methodenaufrufe durchführt. Die Kommunikation zwischen Stub und Skeleton erfolgt über Sockets. Leider wird der Port, auf dem kommuniziert wird, *dynamisch* festgelegt und ändert sich bei jeder neuen Instanzierung. Die Änderung der Portnummern macht es unmöglich, Regeln für die Firewall aufzusetzen, die dem gerecht werden.<sup>2</sup>

---

<sup>2</sup>Für die statische Telepathologie wurde der Server in einen Bereich vor die Firewall ge-

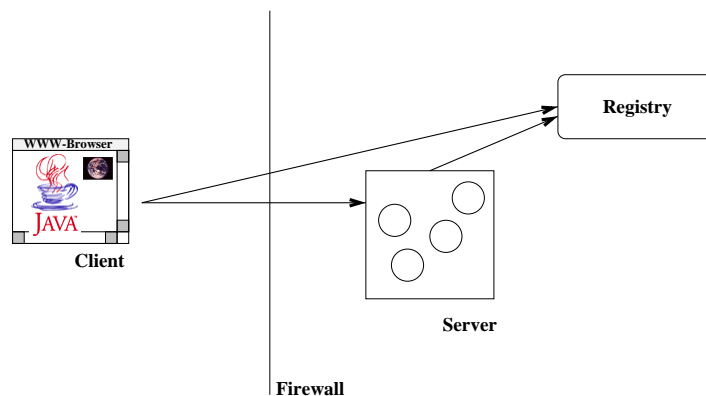


Abbildung 5.6: *Java RMI: Der Server meldet sich bei der Registry an, der Client erhält von der Registry eine Referenz auf den Server. Leider erfolgt die Kommunikation zwischen Client und Server auf dynamisch ermittelten Portadressen.*

CORBA verhält sich an dieser Stelle anders. Der Port für die Kommunikation zwischen Skeleton und Stub kann festgelegt werden und erlaubt die Öffnung dieses Ports in der Firewall. Wie bereits auf S. 40 beschrieben, gestattet CORBA den Aufruf von Objekten, die in einem Netzwerk auf unterschiedliche Computer verteilt sind und in unterschiedlichen Sprachen implementiert sein können (C/C++, Java, COBOL). Ein CORBA-Objekt stellt die in Anhang D dokumentierten IDL-Funktionen bereit, die von dem Java Applet zur Steuerung des Mikroskops verwendet werden. Im einzelnen sind dies Funktionen für die Ansteuerung des x,y-Tisches, Lampe, Objektrevolvers und ein Autofokus.

### 5.3.3 Autofokus

Typischerweise befinden sich die für eine Begutachtung relevanten Zellen in unterschiedlichen Tiefen eines histologischen Präparats und müssen durch geeignete Fokussierung sichtbar gemacht werden. In Abb. 5.7 ist ein schematischer Querschnitt durch ein vollständiges Präparat, wie es in der Praxis verwendet wird, dargestellt.

---

stellt und durch eine eigene Firewall abgesichert. Eine solche Lösung kommt leider für die dynamische Telepathologie nicht in Betracht, da das Mikroskop und der Steuerungsrechner

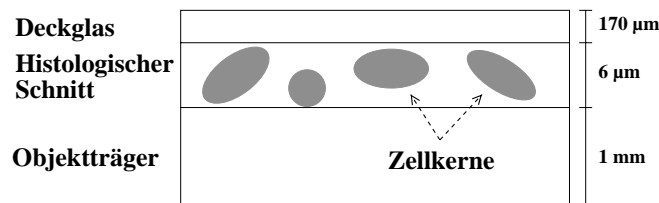


Abbildung 5.7: *Schematischer Querschnitt durch ein Präparat.*

Leider stellt das Mikroskop keinen integrierten Hardware-Autofokus zur Verfügung; mit einem Autofokus lassen sich Fokussierungen sehr viel schneller vornehmen, als wenn für jede Fokusänderung ein Bild zur Überprüfung an den Client übertragen werden muß. Erfahrungsgemäß müßten ca. 20–50 Bilder über das Netzwerk zum Client übertragen werden.

Innerhalb der Abteilung haben Kunz und Schröder in ihren Arbeiten Autofokusverfahren entwickelt, die den Betrag des Gradienten global über ein mit einer Videokamera aufgenommenes Bild maximieren [Kunz 92]. Die Implementierung von Schröder zeigt dabei für die Verwendung in der Tele-diagnostik zwei Schwächen:

- Es wird global optimiert, d.h. es wird eine Fokusposition gefunden, die für das *ganze* Bild optimal ist, aber nicht unbedingt für lokale Regionen.
- Die Implementierung benötigt für einen Fokusvorgang 90 Sekunden. Diese Zeit ist für automatisierte, zeitgesteuerte Aufnahmeverfahren, für die es ja implementiert wurde, ausreichend, nicht aber für die Tele-diagnostik.

Es mußte deshalb ein Verfahren implementiert werden, das innerhalb kurzer Zeit auf kleine Regionen fokussieren kann. Das entwickelte Verfahren basiert auf der Tatsache, daß eine Defokussierung einer Glättung des Bildes entspricht, d.h. hohe Frequenzen im Fourierraum unterdrückt werden. Als Maß für die Fokussierung/Defokussierung wird deshalb das Integral über den hohen Frequenzanteil verwendet. Dieses Maß wird zur Fokussierung in Abhängigkeit von  $z$  optimiert.

---

in den Räumen der AG-KT bleiben müssen.

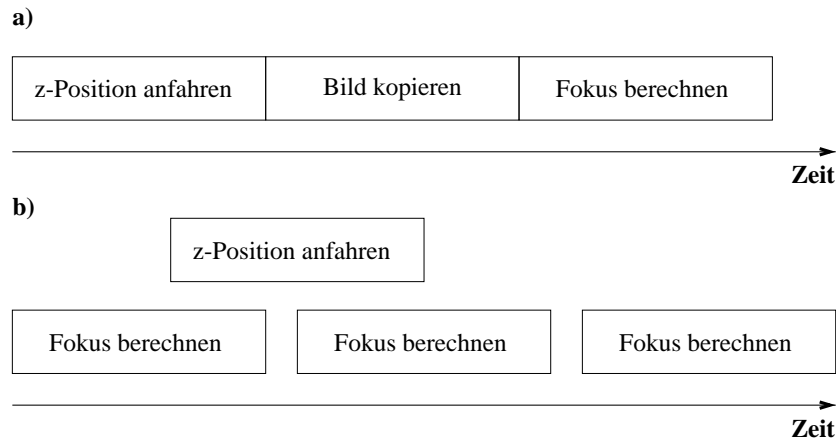


Abbildung 5.8: Zeitdiagramm zum Autofokus. In dem von Schröder implementierten Verfahren werden alle Schritte sequentiell abgearbeitet (a). In dem für die dynamische Telepathologie verwendeten Verfahren erfolgt die Messung der Fokuszahl parallel zur Ansteuerung der z-Position (b).

Die im Steuerrechner eingebaute Framegrabberkarte liest Bilder mit einer Rate von 20 Hz aus und schreibt diese per DMA in den Hauptspeicher. Ein Grund für die niedrige Geschwindigkeit des von Schröder verwendeten Verfahrens lag in der Implementierung: Die Berechnung des Gradienten wurde in Java durchgeführt, ein großer Teil der Rechenzeit wurde mit dem Kopieren des Bildes und mit dem Konvertieren verbracht. Dabei wurde erst das Mikroskop in die gewünschte z-Position gebracht, gewartet bis der Framegrabber ein Bild ausgelesen hatte, das Bild kopiert und zuletzt mit Java ein Maß für den Fokus berechnet. Eine solche Vorgehensweise ist sehr langsam: Das Mikroskop benötigt eine vergleichsweise lange Zeit, um eine bestimmte z-Position anzufahren. Dabei nähert es sich dieser Position zuerst mit großer Geschwindigkeit, verwendet dann aber für die letzten Schritte des Schrittmotors eine sehr niedrige Geschwindigkeit, um eine hohe Genauigkeit zu erreichen. Erst wenn die Position erreicht ist, kann ein Bild für die Fokussierung aufgenommen werden.

Der für die Telediagnostik implementierte Algorithmus verfolgt eine andere Strategie: Für jedes vom Framegrabber aufgenommene Bild wird eine Fokuszahl berechnet. Damit werden Berechnung des Fokusmaßes und Positionierung entkoppelt; sobald das Mikroskop die Zielposition erreicht hat,

steht das Fokusmaß zur Verfügung. Aufgrund der Entkoppelung dieser Prozesse kann es vorkommen, daß das Fokusmaß für eine Position nur in der Nähe der gewünschten z-Position berechnet wird. In der Praxis haben das Rauschen der CCD-Kamera und der AD-Konverter der Framegrabberkarte größere Effekte, als sie durch die Entkoppelung entstehen.

### Fokuszahl

Für die Fokuszahl wird ein Bildausschnitt von  $64 \times 64$  Pixel in der Mitte des Bildes verwendet. Von diesem Bildausschnitt wird in y-Richtung eine FFT durchgeführt. Als Fokuszahl ergibt sich mit der Fouriertransformierten

$$\hat{G}(k, x) = \sum_{k=0}^{63} G_{\text{Bild}}(k, x) e^{i \frac{2\pi k y}{64}} \quad (5.1)$$

die Fokuszahl durch eine gewichtete Summation über den unteren Frequenzbereich des Powerspektrums.

$$F_{\text{Fokus}} = \sum_{x=0}^{63} \sum_{k=0}^{64/3} k^2 \log(1 + |\hat{G}(k, x)|^2) \quad (5.2)$$

Diese heuristische Fokuszahl liefert in der Praxis gute Ergebnisse. Die Geschwindigkeit des eingesetzten Servers reicht aus, um zu jedem Bild des Framegrabbers eine Fokuszahl zu berechnen. Abbildung 5.9 zeigt den Verlauf Fokuszahl/z. Es zeigt sich ein deutlicher Peak um den optimalen Fokuswert herum. Das Rauschen des Bildaufnahmesystems führt zu einer Streuung des Fokusmaßes.

Das implementierte Autofokusverfahren benötigt für eine Fokussierung etwa 3 Sekunden. Durch weitere Optimierungen ließe sich die Zeit nur noch geringfügig reduzieren, da die Geschwindigkeit der Schrittmotoren an dieser Stelle dominiert.

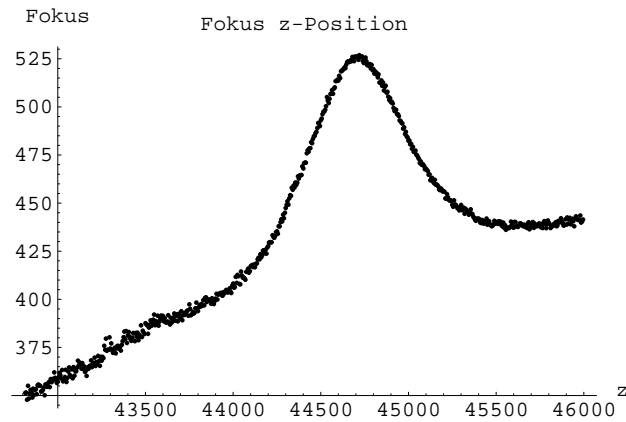


Abbildung 5.9: Verlauf Fokuszahl/z-Position.

### 5.3.4 Bildaufnahme

In Abschnitt 5.1 wurden zwei Strategien zur Bildaufnahme diskutiert. Für die semi-dynamische Telediagnostik werden die mit der Framegrabberkarte aufgenommenen Bilder mit dem JPEG-Verfahren komprimiert und per HTTP an den Client übertragen. Der Client setzt dann mehrere Bilder zu einem Bild zusammen und zeigt das zusammengesetzte Bild an.

Sun stellt für die Aufnahme, Übertragung und Darstellung von Live-Bildern innerhalb von Java Applikationen/Applets das Java Media Framework (JMF) in der Version 2.0 bereit. Leider war das JMF 2.0 zum Zeitpunkt der Entwicklung noch nicht für Linux verfügbar. Als Alternative wurden deshalb Gif-Bilder verwendet: Das Gif-Format erlaubt das Aneinanderhängen von mehreren Gif-Bildern, die dann als Film abgespielt werden. Der Client richtet eine Anfrage an den HTTP-Server, der dann so lange Bilder von der Framegrabberkarte ausliest und an den Client als Gif-Sequenz schickt, wie die Verbindung zum Client nicht geschlossen wird. Die Gif-Bilder werden vom Server erzeugt und per HTTP an den Client übertragen.

Im Gegensatz zu ISDN-Verbindungen ist die Bandbreite bei Internetverbindungen nicht garantiert, sie ändert sich kontinuierlich. Daher paßt sich die Bildwiederholrate an die verfügbare Netzwerkkapazität an. Im lokalen Netzwerk werden etwa 7 Hz erreicht.

### 5.3.5 Benutzeroberfläche

Die dynamische Telediagnostik ist wie bereits die statische Telediagnostik in das vorgestellte Applet-Framework eingebunden. Abbildung 5.10 zeigt die Benutzeroberfläche der DynTed-Komponente. Unterschiedliche Buttons dienen der Änderung von Fokus, Beleuchtung und Objektiv. Der angezeigte Bildausschnitt lässt sich durch Ziehen mit der Maus verschieben. Mit der rechten Maustaste wird ein Kontextmenü angezeigt, in dem auf eine bestimmte Stelle fokussiert, eine bestimmte Position angefahren oder das Objektiv gewechselt werden kann.

Es gibt zwei Möglichkeiten, das Mikroskop zu steuern: Entweder lokal direkt am Mikroskop oder über das Java Applet. Die Umschaltung zwischen diese beiden Modi erfolgt über den Button *Joystick: on/off*. Mit dem Button *Video* wird ein Live-Videobild angezeigt, um z.B. eine manuelle Fokussierung durchzuführen.

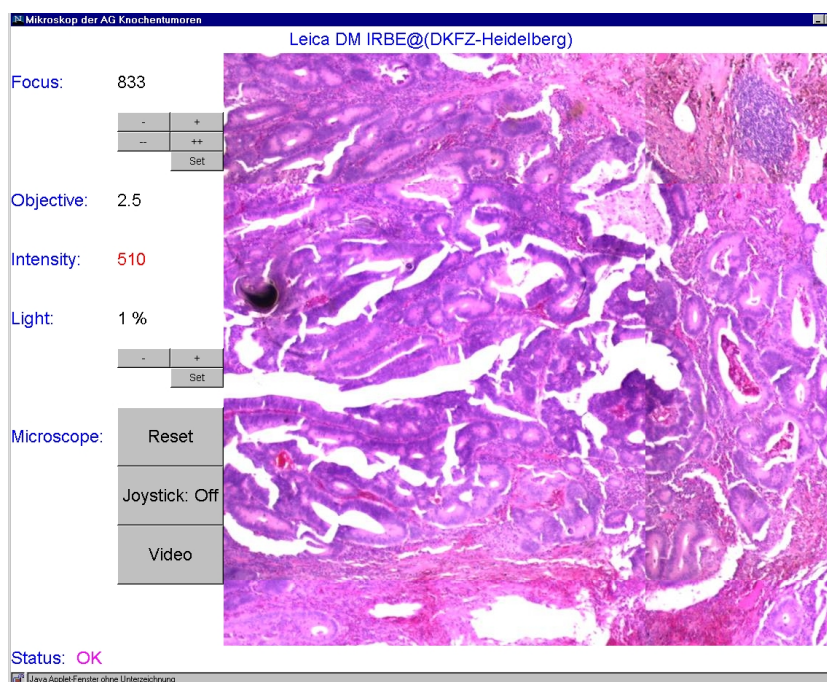


Abbildung 5.10: Benutzeroberfläche zur dynamischen Telepathologie.

### 5.3.6 Steuerrechner

Die DynTed-Komponente für dynamische Telediagnostik besteht aus Client, Steuerungsrechner und motorisiertem Mikroskop. Das Mikroskop und die Videokamera sind am Steuerungsrechner angeschlossen. Als Steuerungsrechner wird ein Linuxsystem mit zwei Pentium-Prozessoren (200 MHz) und einer Matrox Meteor Framegrabberkarte für die Bildaufnahme eingesetzt. Die Ansteuerung des Mikroskops erfolgt über zwei serielle Anschlüsse, einem für den x-y Tisch und einem für Licht, z-Position und Objektivwechsel des Leica DM IRBE Mikroskops.

Der Steuerungsrechner stellt ein CORBA-Objekt für die Steuerung der Mikroskopfunktionen (x,y,z-Position, Licht, Objektiv) und zwei HTTP-Server (Einzelbild/Gif-Sequenzen) zum Abruf der Bilddaten bereit. Das Bildaufnahmemodul speichert die mit der Framegrabberkarte aufgenommenen Bilder als JPEG-Dateien, so daß der Client sie vom HTTP-Server abrufen kann, oder schickt die Bilddaten über einen integrierten HTTP-Server als Gif-Sequenz an den Webbrowser. Dabei paßt sich die Bildaufnahme an die verfügbare Netzbandbreite an.

## 5.4 Zusammenfassung

Es wurde ein System für dynamische Telediagnostik vorgestellt, das als erstes seiner Art komplett webbasiert ist. Es gestattet die Fernsteuerung aller wesentlichen Funktionen eines Mikroskops über das Web. Für die Bildaufnahme stehen zwei unterschiedliche Modi zur Verfügung: ein dynamischer Modus und ein semi-dynamischer Modus. Der semi-dynamische Modus eignet sich gut für Bildübersichten bei niedriger Vergrößerung. Es werden mehrere Bilder der Videokamera zu einem Gesamtbild zusammengesetzt, damit kann das Gesamtbild eine beliebige Größe erreichen. Der dynamische Modus überträgt ein Live-Bild, dabei wird die Bildwiederholrate an die verfügbare Netzwerkbandbreite angepaßt.



# Kapitel 6

## Quantitative Telediagnostik (QuanTed)

In den letzten zwei Kapiteln wurden die Komponenten für die webbasierte statische und quantitative Telediagnostik vorgestellt, bei denen ein Konsultant eine Befundung aufgrund von klinischen Daten und Bilddaten erstellt und dem Referenten die Diagnose mitteilt. Gegenstand dieses Kapitels ist die Komponente QuanTed (Quantitative Telediagnostik), die mit Methoden der digitalen Bildverarbeitung charakteristische Merkmale aus histologischen Bildern extrahiert.

In vielen Bereichen der Medizin haben quantitative Merkmale zur Beschreibung von Bilddaten eine zunehmende Bedeutung für die Diagnose gewonnen. Die Messung von Objekteigenschaften dient z.B. der frühzeitigen Erkennung von Abweichungen von der Norm oder einer objektiveren Bewertung einer Läsion. Ein Beispiel hierfür ist in der Pathologie die Charakterisierung von Gewebestrukturen durch Parameter wie z.B. die Zellmorphologie (Form, Struktur, Verteilung, Häufigkeit) oder molekularbiologischen Parametern (Aktivität von Genen) .

Die Analyse von Tumormaterial mit computergestützten Bildanalyseverfahren in Kombination mit modernen Mikroskopierv Verfahren stellt für den Pathologen eine wertvolle Ergänzung dar [Wootton 95]. Sie ist dem visuellen System des Menschen hinsichtlich quantitativer Beschreibung, Objektivität,

Reproduzierbarkeit und erfassbarer Datenmenge überlegen. Die mit der digitalen Bildverarbeitung ermittelten Parameter tragen zu einer größeren Objektivität bei der Beurteilung von Tumorerkrankungen bei. Beispiele für die Bestimmung quantitativer Änderungen des Gewebes sind:

**Zellmorphologie:** Die Struktur von Zellen, d.h. Form, Größe und Chromatinregionen weisen bei Tumorgewebe andere Häufigkeitsverteilungen auf [Janson 96].

**Genaktivitäten:** Das Gen H19 zeigt beim menschlichen Harnblasenkarzinom eine deutlich höhere Aktivität als bei gesundem Gewebe. Diese läßt sich mit molekularbiologischen Methoden nachweisen und kann für ein computergestütztes Grading verwendet werden [Fischer 96].

**Zytoskelett:** Das Zytoskelett, das als Stützgerüst und Bewegungsapparat des Zellkerns dient, hat bei Tumorgewebe eine andere Struktur und Zusammensetzung. Die entsprechenden Veränderungen können quantitativ gemessen werden [Kattit 99].

Methoden für die Extraktion charakteristischer Merkmale aus histologischen Bildern werden nur an wenigen Forschungszentren entwickelt. Die webgestützte quantitative Telediagnostik bietet die Chance, zu einer größeren Verbreitung quantitativer Analysverfahren beizutragen: Nach geeigneter Präparierung durch ein histologisches Labor werden digitale Bilder des Präparats für eine quantitative Auswertung an einen Server für quantitative Telepathologie verschickt. Der Server führt mittels digitaler Bildverarbeitung eine Analyse durch und stellt die Ergebnisse zum Abruf zur Verfügung.

Bisher gibt es noch keine Telemedizin-Systeme, die eine Komponente für die Ermittlung charakteristischer Merkmale aus histologischen Bildern beinhalten. Das in diesem Kapitel vorgestellte QuanTed-System (Quantitative Telediagnostik) ist das erste, das Bildverarbeitungsdienste für die Pathologie über das Web anbietet.

Die quantitative Telediagnostik verfolgt einen grundsätzlich anderen Ansatz als die statische und dynamische Telediagnostik. Während die statische

und die dynamische Telediagnostik durch eine *Person-to-Person* Konsultation zwischen Konsultant und Referent charakterisiert werden, handelt es sich bei der quantitativen Telepathologie um einen sogenannten *E-Service*. Unter E-Services werden elektronische Dienste verstanden, die ein Benutzer über das Web nutzen kann, ohne daß eine Person direkt in der Erbringung dieser Dienste involviert ist.

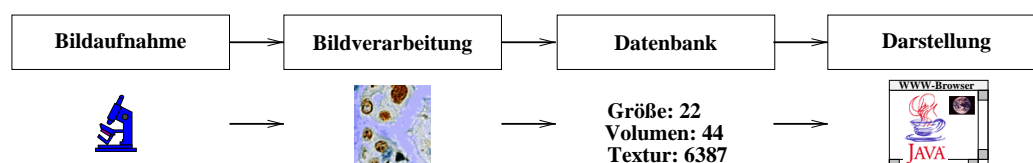


Abbildung 6.1: *Quantitative Telepathologie: Die Bilddaten werden mit StaTed, dem Applet für die statische Telediagnostik, aufgenommen und an den Server geschickt. Per digitaler Bildverarbeitung werden die Bilddaten auf dem Server gefiltert, Objekte segmentiert und Vermessen. Die Ergebnisse werden in der Datenbank gespeichert und können mit einem Webbrowser vom Server abgerufen werden.*

Beim QuanTed-System wird die Ermittlung quantitativer Parametern aus histologischen Bildern als E-Service angeboten (Abb. 6.1). Dazu wird das in Kapitel 4 vorgestellte Applet für statische Telepathologie um eine Komponente für die *Bildverarbeitung und Parameterextraktion* und um eine Komponente zur *Darstellung der Ergebnisse (Report)* erweitert.

## Bildverarbeitung und Parameterextraktion

Aus histologischen Bildern werden durch eine Komposition verschiedener Bildverarbeitungsoperationen, wie z.B. Filterung, morphologische Operationen, Segmentierung etc., Objekte detektiert und vermessen. Bei den Bildverarbeitungsoperationen handelt es sich z.T. um Standardverfahren wie z.B. die Medianfilterung und zum Teil um speziell für Histologie entworfene Verfahren, wie z.B. für die Vermessung von Parametern. Eine Übersicht über die wichtigsten Bildverarbeitungsoperationen des QuanTed-Systems findet sich in Anhang B.

Die Komposition bestehender Operationen zu einem Verarbeitungsver-

fahren kann entweder textuell, d.h. mit einer Hochsprache wie z.B. C++, oder aber grafisch beschrieben werden. Textuelle Darstellungen sind i.a. schwieriger zu entwickeln, nachzuvollziehen und zu warten [Schiffer 98].

Die grafische Beschreibung stellt den Datenfluß zwischen den einzelnen Bildverarbeitungsoperationen intuitiv dar und ist deshalb einfacher zu verstehen, verliert allerdings bei komplexen Problemen an Übersichtlichkeit. QuanTed verwendet aufgrund der intuitiveren Handhabung einen visuellen Programmieransatz für die Entwicklung von Bildverarbeitungsverfahren, eine ausführliche Begründung wird in Abschnitt 6.3.2 gegeben.

Mit der im QuanTed-System integrierten visuellen Programmierungsumgebung lassen sich auf der Basis existierender Bildverarbeitungsoperationen neue Verfahren für die Ermittlung quantitativer Parameter entwickeln. Die Entwicklung solcher Methoden wird von Bildverarbeitungsspezialisten vorgenommen.

Die Anwendung der Verfahren erfolgt durch einen Pathologen. Um die Anwendung so einfach wie möglich zu gestalten, werden einmal entwickelte Verfahren in einer Methodendatenbank abgelegt und lassen sich in einem Dialog auswählen und auf die in der Datenbank befindliche Datensätzen anwenden. Es sind dadurch weder spezielle Kenntnisse der Bildverarbeitung, noch Erfahrungen mit der visuellen Programmierung erforderlich. Die Ergebnisse der Verarbeitung werden in der Datenbank gespeichert und werden mit einem Webbrowser als Report vom Server abgerufen.

Die Bildverarbeitung ist aus Performace-Gründen als Client/Server-System ausgelegt. Die Verarbeitung der Bilddaten erfolgt auf dem Server, der Client steuert die Verarbeitung auf dem Server. Die Bildverarbeitung wird in Abschnitt 6.4 vorgestellt.

### **Darstellung der Ergebnisse (Report)**

Der Bildverarbeitungsschritt speichert die Ergebnisse in Form von Bildern und numerischen Daten in der Datenbank. Ziel der Darstellung der Ergebnisse ist es, dem Pathologen eine möglichst umfassende Übersicht über die ermittelten quantitativen Parameter zu geben. Deshalb werden die Daten als

Report aufbereitet, in dem die Zusammenhänge zwischen Bilddaten, Ergebnissen der Einzelschritte der Bildverarbeitung und den quantitativen Daten dargestellt werden. Anhand des Reports ist es möglich, eventuelle Fehler in der Bildverarbeitung, z.B. aufgrund einer falschen Segmentierung, visuell zu erkennen und die Qualität der ermittelten Daten einzuschätzen.

## 6.1 Aufbau des Kapitels

Nach einigen einleitenden Bemerkungen zur Motivation der quantitativen Telediagnostik im ersten Teil des Kapitels, wird als nächstes eine Übersicht über das QuanTed-System gegeben. Im Anschluß daran wird ein vom Ansatz her ähnliches System für die Zytologie vorgestellt.

Der wichtigste Teil der QuanTed-Komponente bildet die visuelle Programmierung. Deren Ursprung, die Vor- und Nachteile und verfügbare Systeme werden in Abschnitt 6.3.2 diskutiert. Es wird begründet, weshalb bereits erhältliche Systeme ungeeignet sind und es wird die visuelle Programmierungsumgebung des QuanTed-Systems, der Structure Designer, vorgestellt. Den Abschluß dieses Kapitels bilden einige Anwendungen der quantitativen Telediagnostik.

## 6.2 Übersicht QuanTed

Abbildung 6.2 zeigt die Benutzeroberfläche für die quantitative Telediagnostik: Oben links ist das aus der statischen Telepathologie bekannte Applet, rechts daneben die visuelle Programmierungsumgebung, der Structure Designer, zu sehen. Im Structure Designer ist jede Komponente, die Operationen auf einem Bild durchführt, durch einen sogenannten Glyphen dargestellt. Jeder Glyph repräsentiert sich durch eine quadratische Fläche, auf deren linker Seite sich die Eingabe- und auf derer rechten Seite sich die Ausgabekanäle befinden.

Glyphen werden aus einer Liste (oberer Teil des Fensters) ausgewählt und auf dem Playground (unterer Teil des Fensters) mit anderen Glyphen nach den Erfordernissen des Bildverarbeitungsalgorithmus miteinander ver-

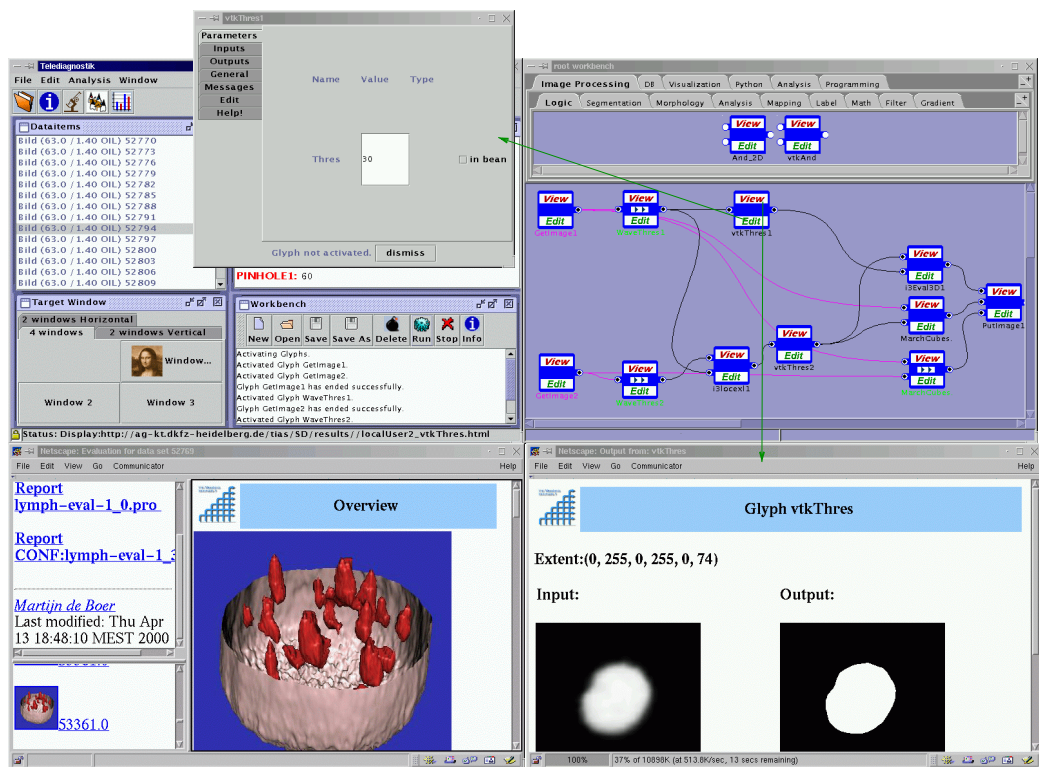


Abbildung 6.2: Quantitative Telepathologie: Oben links: Applet für statische Telepathologie. Oben rechts: Structure Designer. Jede Bildverarbeitungsoperation ist durch ein grafisches Symbol, einen Glyphen, dargestellt. Glyphen werden entsprechend dem gewünschten Datenfluß untereinander verbunden. Zusätzliche Parameter zu einem Glyphen werden im Parameter-Window eingestellt. Unten links: Report mit Ergebnissen einer Auswertung. Unten rechts: Ergebnisse eines Glyphen.

bunden. Zu jedem Glyphen lassen sich in einem Fenster Parameter spezifizieren, etwa die Größe der Filtermaske eines Bildverarbeitungsfilters. Zur Ausführung eines Glyphen wird eine Anfrage an den Server geschickt. Die Ergebnisse eines einzelnen Glyphen werden durch Drücken des View Buttons in einem Fenster des Webbrowsers angezeigt (unten rechts).

Das Applet für statische Telepathologie wurde um ein Menü *Analysis* erweitert, in dem diejenigen Befehle untergebracht sind, um Datensätze auszuwerten, Ergebnisse abzurufen oder neue Verfahren zu entwickeln (Abb. 6.2). Im einzelnen sind dies:

**Workbench:** Mit diesem Menübefehl wird der Structure Designer aufgerufen. In einem Fenster wird die Workbench angezeigt, mit der visuelle Programme geladen, gespeichert und gelöscht werden können. Der Structure Designer mit dem eigentlichen visuellen Programm erscheint in einem separaten Fenster.

**Apply:** Mit dem Structure Designer entwickelte Verfahren lassen sich in der Methoden-Datenbank speichern und mit Apply auf einen Datensatz für die Auswertung anwenden. Nach der Auswahl des gewünschten Verfahrens wird dem Benutzer mit einem Dialog eine Bestätigung angezeigt. Daraufhin wird das Verfahren auf dem Server auf die ausgewählten Bilder angewendet und die Ergebnisse in die Datenbank gespeichert.

**Report:** Die Ergebnisse der Auswertungen werden mit dem Report-Generator erzeugt und im Webbrowser angezeigt (unten links in Abb. 6.2). Die Webseite enthält neben den Ergebnissen, Bilder wesentlicher Zwischenschritte (z.B. Segmentierung), um die korrekte Durchführung der Auswertung überprüfen zu können.

## 6.3 Stand der Forschung

Webbasierte E-Services, d.h. elektronische Dienste, die geleistet werden, ohne daß eine Person direkt in ihrer Erbringung involviert ist, sind ein sehr junges Konzept. Erst die Verfügbarkeit von Java bietet Mechanismen zur Realisierung hochinteraktiver webbasierter E-Services. Dementsprechend ist auf dem Gebiet der E-Services bisher wenig entwickelt worden. Als Vorläufer von komplexen E-Services, wie sie mit dem WebTed-System möglich sind, wird in Abschnitt 6.3.1 das TELEQUANT-System vorgestellt.

Die visuelle Programmierung hat einen großen Anteil an dem hier vorgestellten Ansatz für quantitative Telepathologie. Obwohl die Konzepte für visuelle Programmiersprachen bereits seit den 60er Jahren bekannt sind, haben sie sich nur eingeschränkt durchsetzen können. Der Stand der Forschung und das Pro und Kontra der visuellen Programmierung werden in Abschnitt 6.3.2 diskutiert.

### 6.3.1 Telequantisierung

Als Vorläufer der quantitativen Telepathologie kann das an der Uni Dresden entwickelte TELEQUANT-System für die Zytometrie betrachtet werden. Es wurde im Rahmen des EUROPATH-Projekts der EU entwickelt, um Ergebnisse des Zellkern-DNA-Gehalts zu vergleichen, die von unterschiedlichen kommerziell verfügbaren Systemen ermittelt wurden [Kunze 98]. Kommerziell erhältliche Systeme unterscheiden sich in den verwendeten Verfahren für die Bestimmung des DNA-Gehalts, so daß die Ergebnisse untereinander nicht unmittelbar vergleichbar sind. Benutzer können die Ergebnisse ihres Systems als ASCII-Textdaten per Mail an den Server schicken; die Ergebnisse werden dann mit standardisierten Algorithmen und statistischen Tests ausgewertet und die Ergebnisse dem Benutzer präsentiert.

### Bewertung

Das Dresdener System ist auf eine spezielle Anwendung zugeschnitten, nämlich den Vergleich der Ergebnisse des Zellkern-DNA-Gehalts zwischen unterschiedlichen Systemen. Es erlaubt keine Interaktionen, ist nicht in ein Telepathologiesystem integriert und wurde durch dynamisch erzeugte HTML-Seiten mit cgi-bin Skripten realisiert. Das TELEQUANT-System ist von den Fähigkeiten und von der Komplexität her sehr viel einfacher als das hier vorgestellte System zur allgemeinen Bildverarbeitung.

### 6.3.2 Visual Programming

*Ein Algorithmus muß gesehen werden, damit er begreifbar wird,  
und der beste Weg zu lernen, wie ein Algorithmus funktioniert,  
ist, ihn auszuprobieren.*

DONALD KNUTH

Diese Worte sind dem Klassiker *The Art of Computer Programming* entnommen [Knuth 73]. Schon sehr früh wurden grafische Darstellungen von



Computerprogrammen dazu verwendet, das Verständnis der Zusammenhänge eines Computerprogramms zu fördern. Bereits Ende der 50er Jahre existierten Programme, um automatisch aus Assembler und Fortran Programmen Flußdiagramme zu erstellen (vergl. [Price 93]). Spätere Systeme gingen den umgekehrten Weg und verwendeten grafische Beschreibungen, um daraus Sourcecode zu generieren. Am IBM Forschungslabor in San Jose wurde das *Programming Support System* entwickelt, mit dem aus Struktogrammen (Nassi-Schneidermann-Diagramme) PL/I Code erstellt wurde [Frei 78].

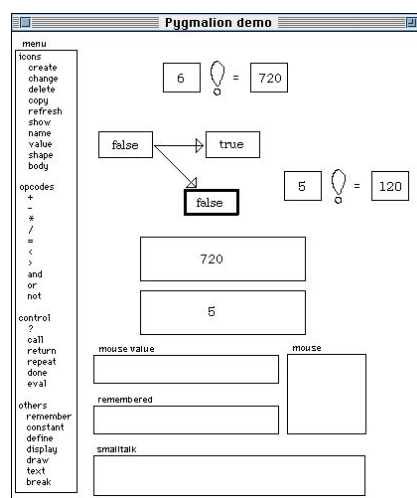


Abbildung 6.3: Simulation des Pygmalion Systems mit Apple Hypercard von Allen Cypher.

Als richtungsweisend gilt das 1975 von David C. Smith im Rahmen seiner Dissertation entwickelte System *Pygmalion* [Smith 75]. Dieses System entstand aus der Überlegung heraus, daß Programme anhand von relevanten Beispielen besser verstanden werden und daß sich Piktogramme gut zur Darstellung von Kontrollstrukturen eignen. Das Pygmalion-System integrierte eine rein visuelle Programmiersprache, d.h. ein Programm wurde durch die Anordnung von Piktogrammen grafisch erstellt. Obwohl nur für einfache Aufgaben, wie z.B. der Berechnung von  $n!$  geeignet, galt Pygmalion als Vorbote grafischer Benutzerschnittstellen.

Mit der Verfügbarkeit von Rechnern mit grafischen Benutzeroberflächen in den 80er Jahren verstärkten sich auch die Forschungsaktivitäten hinsicht-

lich visueller Programmierung. Ein Großteil der Forschung zur visuellen Programmierung wurde in den letzten 15 Jahren durchgeführt. Man hatte die Vision, durch visuelle Programmierung zum einen die Softwareentwicklung zu beschleunigen und zum anderen Anwendern ohne Kenntnisse textueller Sprachen die Erstellung und Anpassung von Software zu ermöglichen. Dieses Ziel wurde in verschiedenen Ansätzen verfolgt.

*Programming by Demonstration (PbD)* ist eine Technik zur Automatisierung von Operationen bei denen der Benutzer einige Operationen ausführt, um sie zu verallgemeinern und später mit anderen Daten zu wiederholen. Eine Übersicht über verfügbare PbD-Systeme findet sich in [Allen 93].

Das *Control Flow* Paradigma verwendet Piktogramme, um den Kontrollfluß eines Programms zu steuern. Operationen, wie Zuweisungen oder Funktionen, sind durch quadratische Piktogramme dargestellt, die mit Linien verbunden werden, um den Kontrollfluß darzustellen. Der Kontrollfluß kann durch Kontrollkonstrukte wie Schleifen oder Abfragen gelenkt werden. Das Control Flow Paradigma kann als grafische Darstellung (d.h. Flußdiagramm) einer textuellen, prozeduralen Sprache betrachtet werden. Ein Beispiel dafür ist das System Pict [Glinert 84].

Das *Data Flow* Paradigma verwendet grafische Piktogramme, die durch Linien miteinander verbunden sind und den Datenfluß repräsentieren. Jedes Piktogramm entspricht einer Operation, die auf den Datenfluß angewendet wird. Das Data Flow Paradigma läßt sich mit funktionalen Sprachen vergleichen. Beispiele sind etwa Show-and-Tell, Khoros und IBM DX [Kimura 86] [Khoros] [OpenDX].

Das *visual rewriting* Paradigma stellt Zusammenhänge, z.B. Relationen zwischen Entitäten, grafisch dar und wendet Regeln an, um Teile eines Graphen durch Ersetzung zu transformieren (*graph rewriting*). Beispiele für die visuelle Programmierung von Datenbankabfragen sind z.B. [Rodgers 97] [Schurr 94] [Gyssens 94].

Die unterschiedlichen dargestellten Ansätze haben der visuellen Programmierung nicht zum großen Durchbruch verholfen; dies hat mehrere Ursachen:

**Skalierbarkeit:** Visuelle Programme benötigen im Vergleich zu textuell be-

schriebenen Programmen mehr Platz zur Darstellung. Bei komplexen Programmen ist die visuelle Darstellung nicht mehr überblickbar und der Vorteil geht dadurch verloren. Neuere Ansätze versuchen dem durch 3D-Darstellungen (z.B. Cube) oder die effizientere Nutzung des Bildschirms beizukommen [Najork 91] [Citrin 96].

**Standards:** Es haben sich für visuelle Programmierung keine Standards etabliert. Um ein visuelles Programm verstehen zu können, muß deshalb erst die Notation verstanden werden. Eine visuelle Darstellung von Programmen führt nicht automatisch zu einem besseren Verständnis. Untersuchungen hierzu wurden von J. Good durchgeführt [Good 99].

**Wartung:** Visuelle Programme werden oft nicht im top-down Verfahren erstellt, es findet keine Trennung zwischen Daten, GUI und Funktionalität statt. Eine spätere Erweiterung der Programme wird durch die fehlende Kapselung erschwert. Neuere Ansätze schlagen die Verwendung von CASE-Tools und Erweiterungen in Richtung Objektorientierung vor [Sauer 99] [Braine 97].

Nichtsdestoweniger hat sich der Ansatz der visuellen Programmierung in einigen Nischenbereichen durchgesetzt. Das bekannteste Beispiel hierfür ist wohl LabVIEW von National Instruments, das 1986 auf den Markt kam [LabView]. LabVIEW ist eine Programmierumgebung zur Datenerfassung, Meßgerätesteuerung, Analyse und Darstellung. Mit einer visuellen Programmiersprache können Module zur Ansteuerung von Meßinstrumenten zu einem virtuellen Meßinstrument zusammengesteckt werden, das über eine anspruchsvolle Benutzeroberfläche eine Interaktion mit verschiedenen am Rechner angeschlossenen Geräten erlaubt.

In den Marketingabteilungen der Hersteller von Programmierwerkzeugen ist der Begriff *Visual* inzwischen zum Synonym für einfache Softwareentwicklung geworden, bei der die Programmerstellung durch grafische Interaktionsmöglichkeiten erleichtert wird. Schiffer hat Visuelle Programmierung folgendermaßen definiert [Schiffer 98]:

*Visuelle Programmierung ist jene Art der Programmierung, für die das visuelle Wahrnehmungssystem des Menschen unentbehrlich ist.*

Z.Zt. lassen sich drei Bereiche ausmachen, in denen visuelle Programmierung eingesetzt wird: Entwurf grafischer Benutzeroberflächen (etwa *Visual Basic*, *Visual C++*, *Visual Age*), toolgestütztes Softwaredesign (CASE Tools, Jackson-Diagramme, Datenflußdiagramme) und grafische Programmierung mit Piktogrammen, die den Daten- oder Kontrollfluß zwischen einzelnen Modulen darstellen (LabView, IBM Data Explorer, AVS, Khoros). In der Domäne der Bildverarbeitung haben sich Systeme durchgesetzt, die das Datenflußparadigma verwenden. Im folgenden wird ein Überblick über verfügbare Systeme für die Bildverarbeitung gegeben.

### 6.3.3 open DX (vormals IBM DX)

Der IBM Data Explorer (DX) ist ein professionelles visual Programming Tool für die Analyse und insbesondere die Visualisierung von 2D und 3D Daten. Existierende Komponenten für Isosurface-Generierung, Visualisierung und Analyse können nach dem Datenflußprinzip zu einem Programm zusammengesteckt werden (Abb. 6.4).

Die Funktionalität läßt sich durch das Hinzufügen neuer Verarbeitungsmodule erweitern, allerdings müssen diese in C oder Fortran geschrieben sein. Obwohl der Data Explorer als Client/Server-System ausgelegt ist, ist er für den Einsatz in einem Web-Umfeld aus zwei Gründen ungeeignet: Zum einen läßt sich die in C geschriebene Benutzeroberfläche nicht in Anwendungen integrieren, die im Webbrowser ablaufen sollen; zum anderen werden keine Sicherheitsmechanismen wie z.B. Verschlüsselung unterstützt [OpenDX].

### 6.3.4 AVS

Advanced Visual System bietet das Programm AVSExpress mit ähnlichen Möglichkeiten zur Datenanalyse und Visualisierung wie DX an. Es ist ebenfalls als Client/Server-System ausgelegt und kann durch Module, die in C/C++

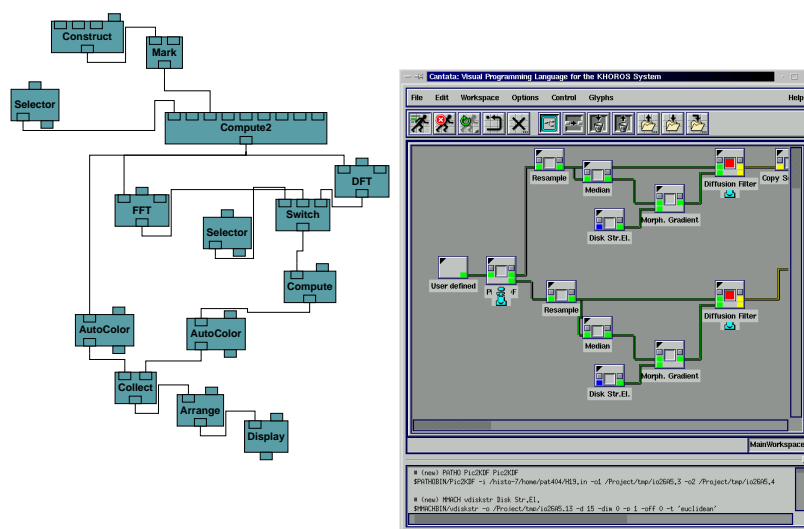


Abbildung 6.4: Links: Netzwerk des IBM Data Explorers. Rechts: Programm zur Analyse von H19-Zellkernen mit Khoros (siehe Abschnitt 6.6.2).

oder Fortran geschrieben sind, erweitert werden. Im Vergleich zu DX können auch grafische Benutzeroberflächen erstellt werden. AVS eignet sich aus ähnlichen Gründen wie bereits DX nicht für den Einsatz im Web [AVS].

### 6.3.5 Khoros

Khoros von Khoros Research ist die Dritte im Bunde der bekannten Entwicklungsumgebungen für Bildverarbeitung. Es ist von den Möglichkeiten her weniger professionell als AVS und DX und bietet im Gegensatz zu AVS keine Möglichkeiten für die Erstellung von Benutzeroberflächen. Im Vergleich zu AVS und DX verfügt Khoros jedoch über Möglichkeiten, alle Arten von Programmen, wie z.B. Shell-Skripte, als Module einzubinden. Auch Khoros kann aus den gleichen Gründen wie AVS nicht im Web eingesetzt werden [Khoros].

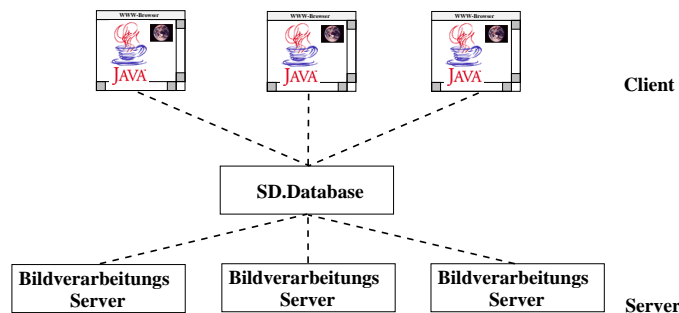
## 6.4 Ansatz zur webbasierten visuellen Programmierung

Die visuelle Programmierung hat sich in der Praxis als gut geeignet für die Entwicklung von Programmen zur digitalen Bildverarbeitung erwiesen. Gründe sind die vergleichsweise niedrige Komplexität, im Vergleich z.B. zur Entwicklung von grafischen Benutzeroberflächen inklusive Programmlogik mit visuellen Programmiersprachen (vergl. [Schiffer 98]).

Existierende Programmpakete haben sich aufgrund der besonderen Sicherheitsanforderungen im Web für die quantitative Telepathologie als ungeeignet erwiesen. Es öffnete sich daher der Weg, eine speziell auf das Web und die Pathologie zugeschnittene Entwicklungsumgebung für Bildverarbeitung zu erstellen.

Dabei konnte auf die Vorarbeiten aus der Diplomarbeit von Wolfgang Römer zurückgegriffen werden [Römer 97]. In seiner Diplomarbeit hat Römer den STRUCTURE DESIGNER (SD), eine visuelle Programmierungsumgebung für die Entwicklung von Genetischen Algorithmen, entwickelt. Der Structure Designer ist ein in Java geschriebenes 3-Tier Client/Server-System und erfüllt die speziellen Sicherheitsanforderungen für den Einsatz im Web. Ein Applet bildet die Benutzerschnittstelle zum System. Der Benutzer meldet sich mit dem Applet bei der zentralen Verwaltungsinstanz (SD.Database) an. Die SD.Database verwaltet die Privilegien der einzelnen Benutzer, die Paßwörter für die Anmeldung und die für den Benutzer verfügbaren Bildverarbeitungsmodule. Am Server kann sich eine beliebige Anzahl von Bildverarbeitungsservern anmelden, welche die eigentliche Ausführung von Programmen übernehmen.

Aufgrund der Plattformunabhängigkeit von Java können die Bildverarbeitungsserver im Netzwerk auf verschiedenen Rechnern laufen, z.B. läuft ein Bildverarbeitungsserver läuft unter Windows NT, ein anderer unter Linux. Die SD.Database ist für die Verteilung der Prozesse auf die einzelnen Rechner zuständig, d.h. sie weiß, welche Programme auf welchen Rechnern verfügbar sind. Aus diesem Grund läuft jegliche Kommunikation zwischen

Abbildung 6.5: *Die Architektur der Structure Designers.*

Client und Bildverarbeitungsserver über die SD.Database.

Leider hatte der Structure Designer einige Einschränkungen, die ihn für den Einsatz in der Bildverarbeitung nur beschränkt nutzbar machten: Es wurde davon ausgegangen, daß jede Operation nur *ein* Eingangs- und *ein* Ausgabedatum verwendet. In der Bildverarbeitung ist es aber üblich, daß Operationen mehr als nur ein Ein- oder Ausgabedatum verwenden. Beispiele sind morphologische Filter oder die Subtraktion zweier Bilder.

Jörg Mattes hat die Arbeit von Wolfgang Römer fortgesetzt und den Structure Designer für den Einsatz in der Bildverarbeitung angepaßt. Es wurde die Netzwerkkommunikation zwischen Client und Server von einem proprietären Protokoll auf RMI umgestellt, das Eventhandling an Java 1.1 angepaßt und die Möglichkeit geschaffen, Programme mit mehreren Ein- und Ausgängen einzusetzen [Mattes 98]. In der von Jörg Mattes fertiggestellten Version werden von den Bildverarbeitungsservern Java Beans Komponenten und Kommandozeilenprogramme unterstützt, d.h. Programme der Form:

```
programm -i1 <in 1> -i2 <in 2> -o1 <out 1> -o2 <out 2> -opt
```

wobei in und out Dateien sind, aus denen Ergebnisse gelesen bzw. in die Ergebnisse geschrieben werden.

## Erweiterungen

Die quantitative Telepathologie ist in das Applet Framework eingebettet. Für eine Verarbeitung der im Archiv gespeicherten Daten sind spezielle Glyphen

für den Zugriff auf die Datenbank notwendig. Für den Einsatz im QuanTed-System wurde der SD deshalb an einigen Punkten erweitert:

**Glyphen:** In der von Jörg Mattes fertiggestellten Version werden fast ausschließlich Kommandozeilenprogramme eingesetzt. Der Datenaustausch zwischen den Programmen erfolgt über Dateien. Bei kleinen Datenmengen, wie sie etwa bei 2D Bildern anfallen, bereitet dies keine Probleme. Bei 3D-Datensätzen müssen erheblich größere Datenmengen verarbeitet werden, so daß andere Kommunikationsformen erstrebenswert sind.

Die ausschließliche Verwendung von Kommandozeilenprogrammen erfordert es, für jeden Glyph ein eigenes Programm zu erstellen. Dadurch gestaltet sich die Einbindung neuer Bildverarbeitungsbibliotheken relativ aufwendig. Der Structure Designer wurde deshalb erweitert, um Bibliotheken, wie z.B. vtk, direkt als Glyph einzubinden ohne vorher dafür ein Kommandozeilenprogramm zu erstellen [VTK].

**Benutzeroberfläche:** In der von Mattes fertiggestellten Version des SD sind alle verwendeten Glyphen *statisch*: Zu jedem Glyph gibt es ein Fenster, um die notwendigen Parameter zu spezifizieren. Dies sind statische Parameter, die ohne weitere Verarbeitung an den Server weitergegeben werden. An einen Glyph angepaßte Benutzeroberflächen hat Mattes nicht vorgesehen, obwohl diese für viele Zwecke durchaus sinnvoll wären, so etwa für die interaktive Markierung von Regionen in Bildern. Es war deshalb notwendig, *dynamische* Glyphen mit der Möglichkeit zur Erstellung von Benutzeroberflächen zu schaffen, die speziell an einen individuellen Glyphen angepaßt sind.

Für die Implementierung des SD wurde von Wolfgang Römer und Jörg Mattes das Abstract Window Toolkit verwendet. Für die Telepathologie werden aufgrund der größeren Möglichkeiten die neueren Java Foundation Classes verwendet. Da diese nur eingeschränkt zueinander kompatibel sind, ist für die Integration in die Telepathologie eine Überarbeitung der Benutzeroberfläche notwendig [JFC Tutorial].

**Darstellung der Zwischenergebnisse:** Khoros, AVS und DX verwenden



spezielle Glyphen, um Zwischenergebnisse anzuzeigen, d.h. jeder Glyph in einem Glyphennetzwerk kann mit einem Viewer-Glyph verbunden werden, um Ergebnisse darzustellen. Im SD ist dies nicht möglich, da Programme sich nur auf dem Bildverarbeitungsserver ausführen lassen. Es wurden daher Mechanismen geschaffen, um Resultate der einzelnen Glyphen im Webbrowser darzustellen.

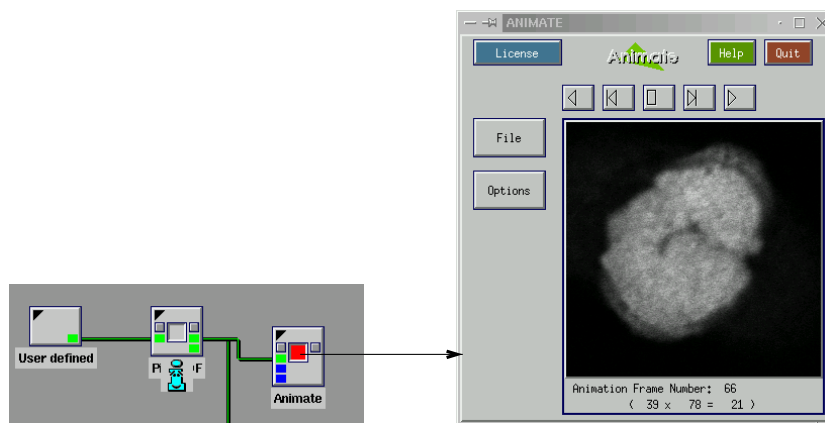


Abbildung 6.6: Um die Ergebnisse eines Glyphen darzustellen, verwendet Khoros spezielle Glyphen. Der Glyph Animate zeigt nacheinander alle Schichten eines 3D Datensatzes an.

## 6.5 Konzept zur Realisierung der webbasierten visuellen Programmierung

Bekannte Systeme wie AVS, DX und Khoros stellen eine API zur Erweiterung der Funktionalität durch neue Glyphen bereit. Sowohl der Client, als auch der Server sind in C/C++ geschrieben, so daß die Bereitstellung einer C/C++-API der günstigste Weg ist. Beim SD ist dieser Ansatz in *der* Form nicht möglich. Grund hierfür ist das bisherige Konzept der strikten Trennung zwischen Server und Client: Die Ausführung von Kommandozeilenprogrammen erfolgt ausschließlich auf dem Server, der Client wird nur für die visuelle Programmierung verwendet. Bisher existierte keine Möglichkeit, Teile des Glyphen clientseitig in Java ablaufen zu lassen.

In den folgenden Abschnitten wird ein Konzept für die einfache und gleichzeitig flexible Erweiterung des SD durch neue Glyphen vorgestellt. Die serverseitige Erweiterung ist in Abschnitt 6.5.1 beschrieben, die clientseitige Erweiterung in Abschnitt 6.5.2.

### 6.5.1 Dynamische Glyphen

Die einfachste Möglichkeit der Integration von Bibliotheken in den SD bestünde in der Bereitstellung einer entsprechenden C++ API mit der notwendigen I/O-Funktionalität und der Verwendung von CORBA für die Kommunikation zwischen den clientseitigen Java-Glyphen und einer C++ Bibliothek auf dem Server. Um einen Glyphen auszuführen, würde der Glyph von einem ORB<sup>1</sup> eine Objektreferenz auf ein Objekt erhalten, das Bildverarbeitungsoperationen bereitstellt. Es könnten mittels der Methoden des Objekts Parameter gesetzt werden, und der Glyph könnte Bildverarbeitungsfunktionen auf dem Server ausführen. Die Kommunikation würde direkt zwischen dem clientseitigen Glyphen und dem serverseitigen CORBA-Objekt erfolgen.

Dieser Ansatz wäre weder besonders innovativ noch besonders elegant: Bibliotheken für Bildverarbeitung mit eigenen APIs gibt es bereits in ausreichender Zahl (z.B. Heurisco, Matrox MIL, VTK, IDL), so daß es unnötig ist, eine weitere zu erstellen. Vielmehr sollte es das Ziel sein, solche Bibliotheken einfach einzubinden und in Glyphen zu verwenden. Im Vergleich zu den existierenden Paketen wie IBM DX, AVS und Khoros ist einer der großen Vorteile des SD die Flexibilität, *beliebige* Komponenten einzubinden. Durch die starre Bindung mit einer API für den Server würde ein großer Teil dieser Flexibilität verloren gehen.

Die Verwendung von CORBA ist außerdem aufwendiger, als es auf den ersten Blick scheint: Abbildung 6.7 zeigt den Weg, um eine Funktion aus einer existierenden Bildverarbeitungsbibliothek mit CORBA in einen Glyphen einzubinden.

Es muß ein Wrapper erstellt werden, der das Bindeglied zwischen der Bibliotheksfunktion und dem Glyphen ist. Der Wrapper übernimmt die Kom-

---

<sup>1</sup>ORB: Object Request Broker

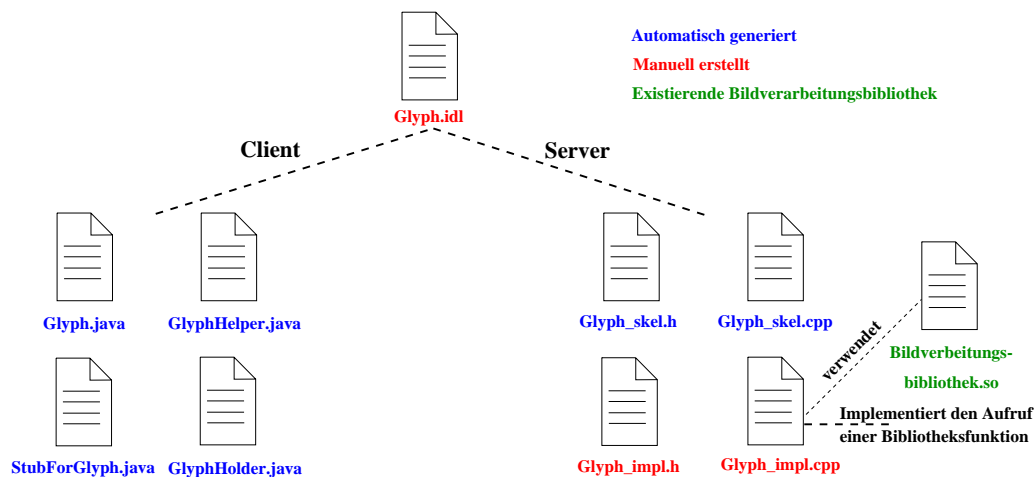


Abbildung 6.7: CORBA Wrapper: Notwendige Dateien.

munikation zwischen clientseitigen Glyphen und Server.

Das Interface des Wrappers wird in IDL definiert (Glyph.idl). Aus dieser Definition generiert der IDL-Compiler die Dateien für die Serverfunktionalität (Glyph\_skel.h, Glyph\_skel.cpp) und die Stub-Dateien (Glyph.java, GlyphHelper.java, StubForGlyph.java, GlyphHolder.java), um die Funktion von einem Glyphen aus aufzurufen. Leider muß die Implementierung mit dem Aufruf der Bibliotheksfunktion *manuell* erstellt werden (Glyph\_impl.h, Glyph\_impl.cpp).

Der hier beschriebene Ansatz wurde in einem Prototyp implementiert, um die Imagemagick-Bibliothek einzubinden. Es zeigten sich zwei Mißstände, die den Ansatz unpraktikabel machen: Die verwendete CORBA Implementierung unterstützte in der damals verwendeten Version keine Shared Libraries, so daß aus einem einfachen CORBA-*Hello World* Programm ein 8 MB Koloß wurde. Dementsprechend schwerfällig ist die Entwicklung. Zudem ist die manuelle Implementierung eines Wrappers sehr zeitaufwendig und fehlerträchtig. Es wurde deshalb ein alternativer Ansatz entwickelt.

#### 6.5.1.1 Ansatz

Seit der Entwicklung der ersten Computer wurden zahlreiche Hochsprachen für deren Programmierung entwickelt. Neue Sprachen versuchten Defizite äl-

terer Sprachen zu beheben, um so Programme effizienter erstellen zu können. Beispiele sind Fortran für die Erstellung numerischer Simulationen, COBOL für die Erstellung betriebswirtschaftlicher Anwendungen oder neuere Sprachen wie C, C++ und Java für allgemeine Programmentwicklung.

Neben diesen Sprachen gibt es noch Skriptsprachen, die bevorzugt für Rapid Prototyping verwendet werden. Skriptsprachen werden interpretiert und weisen einen geringen Grad an Typüberprüfung auf. Kompilierte Hochsprachen wie C++ überprüfen bei der Kompilierung die Zulässigkeit von Operationen mit dem jeweiligen Typ, während diese Überprüfung bei Skriptsprachen zur Laufzeit vorgenommen wird. Diese können daher auf die strenge Typsicherheit verzichten und erhöhen dadurch die Flexibilität. Code kann dynamisch zur Laufzeit erzeugt und mit dem *eval*-Befehl ausgeführt werden. Es ist nicht notwendig, Variablen vor deren Verwendung zu deklarieren oder ihren Typ festzulegen.

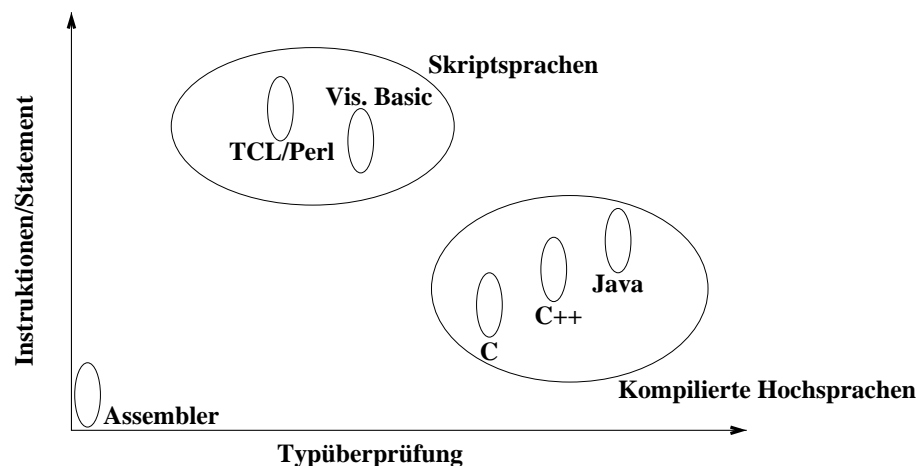


Abbildung 6.8: Für Sprachen mit schwacher Typüberwachung wie Skriptsprachen werden pro Zeile mehr Maschinenbefehle ausgeführt, als für Sprachen mit strenger Typüberwachung (aus [Ousterhout 98]).

Für eine Zeile Skriptsprache mehrere hundert Maschinenbefehle ausgeführt, während für eine Zeile einer kompilierten Hochsprache etwa 5 Befehle ausgeführt werden [Ousterhout 98]. Ein Teil des Mehraufwands resultiert aus der Typüberprüfung zur Laufzeit, ein anderer Teil aus der höheren Funktionalität der Skriptsprachen. Eine Skriptsprache benötigt für ein Programm

weniger Zeilen, als z.B. C++ bei gleicher Funktionalität und ist schneller zu implementieren [Prechelt 00].

In der Praxis werden Skriptsprachen dazu verwendet, um aus Komponenten eine Applikation zu erstellen. Komponenten sind in einer kompilierten Hochsprache geschrieben, so daß in den meisten Fällen alle performance-kritischen Teile in der kompilierten Sprache ablaufen und die Skriptsprache nur verwendet wird, um diese miteinander zu verbinden. Erfahrungen mit der Erweiterung von Khoros mit C++ zeigen, daß sich Skriptsprachen auch gut zur Erstellung neuer Glyphen einsetzen lassen. Das Programm für einen typischen Glyphen besteht aus vier Schritten:

**Einlesen:** Die Eingangsbilder werden eingelesen und zur Verarbeitung vorbereitet. Es werden unterschiedliche Überprüfungen, z.B. der Farbtiefe und der Dimensionalität der Bilder durchgeführt.

**Parameter:** Die angegebenen Parameter werden abgefragt, und es wird überprüft, ob die Angaben sinnvoll waren. Eventuell wird eine Umwandlung der Parameter vorgenommen, z.B. eine Umwandlung einer 0-100% Skala in den Wertebereich 0-255.

**Verarbeitung:** Die Eingangsdaten werden verarbeitet, d.h. die eigentliche Bildverarbeitungsoperation wird durchgeführt. In der Regel wird hierfür eine Funktion aus einer Bildverarbeitungsbibliothek aufgerufen.

**Ergebnisse:** Die Ergebnisse der Verarbeitung werden abgespeichert und stehen dem nächsten Glyphen zur Verfügung.

Von diesen vier Schritten ist nur der Verarbeitungsschritt sehr rechenintensiv. Wenn aber der rechenintensive Teil als C/C++ Bibliotheksfunktion realisiert wird, kann ein Glyph ohne große Performanceverluste mit einer Skriptsprache erstellt werden. Bibliotheksfunktionen lassen sich in nahezu alle Skriptsprachen als Erweiterungsmodule einbinden, um sie innerhalb der Skriptsprache zu verwenden. Die Erstellung von Erweiterungsmodulen läßt sich durch Wrappergeneratoren automatisieren, Details werden im Abschnitt Implementierung besprochen. Damit ergibt sich für Glyphen folgendes Konzept:

*Der SD wird um einen neuen Typ von Glyph erweitert, den PY-SCRIPTGLYPH. Anstelle von Kommandozeilenprogrammen werden Skripte verwendet, die durch einen persistenten Interpreter vom Server ausgeführt werden. Die Verwendung einer Skriptsprache führt nur zu geringen Leistungseinbußen, da geschwindigkeitskritische Teile als Bibliotheksfunktion in C/C++ realisiert werden und als Erweiterungsmodul in die Skriptsprache eingebunden werden. Die Erzeugung der Wrapperfunktionen für die Erweiterungsmodule wird automatisiert.*

*Ein- und Ausgabedaten werden zwischen unterschiedlichen Glyphen entweder mittels Referenzen (innerhalb der Skriptsprache) oder mittels Dateien übergeben.*

#### 6.5.1.2 Implementierung

Die Implementierung verwendet Python als Skriptsprache [Python]. Als interpretierte und objektorientierte Skriptsprache wird Python oft mit Tcl und Perl verglichen, unterscheidet sich aber vor allem durch die einfache und logische Syntax der Sprache. Erfahrene Programmierer können in weniger als zwei Stunden die Konzepte von Python verstehen und anwenden.

Moderne Konzepte wie Module, Klassen, Exceptions und dynamische Typprüfung werden unterstützt. Der Interpreter führt eine Typüberprüfung zur Laufzeit durch und erlaubt die dynamische Ausführung per *eval*-Befehl. Damit kann Code, der erst zur Laufzeit bekannt ist, dynamisch ausgeführt werden. Im folgenden Beispiel werden zwei Zahlen definiert und dann in einer zur Laufzeit definierten Funktion verwendet. Das Skript

```
a=6
b=6
c=eval("a*b+6")
print c
```

liefert als Ergebnis

Diese Eigenschaft wird im SD benötigt: Skripte werden mit dem Client erstellt, über das Netzwerk auf den Server gespeichert, in einen persistenten Interpreter geladen und ausgeführt. Durch die Fähigkeit der dynamischen Ausführung kann ein Skript modifiziert werden, ohne daß es notwendig ist, den Interpreter für eine erneute Ausführung neu zu starten.

Die Verwendung eines persistenten Interpreters bringt zwei Vorteile mit sich: Zwischen einzelnen Glyphen lassen sich in einem Skript instanzierte Objekte durch Referenzen übergeben, d.h. der Datenaustausch zwischen Glyphen mittels Dateien ist nicht mehr notwendig. Jeder Start des Interpreters kostet unter Linux auf einem Pentium 200 etwa 0.2 Sekunden. Durch die Verwendung eines persistenten Interpreters wird also diese Zeit für jeden Glyphen eingespart.

### **Datenformate**

Kommerzielle Systeme verwenden für die Speicherung von Daten ein universelles Datenformat. In dieses Datenformat müssen alle unterschiedlichen Datentypen wie Bilder, Text, Oberflächen gespeichert werden. Der Nachteil dieses Vorgehens besteht darin, daß durch den allgemeinen Ansatz Besonderheiten für einen bestimmten Datentyp nicht berücksichtigt werden. Regelmäßig strukturierte Daten wie z.B. Bitmaps werden effizient über Bildebenen oder Linien adressiert, während dies bei Daten zur Oberflächenbeschreibung nicht der Fall ist [Schroeder 97]. Im Sinne einer größtmöglichen Flexibilität und um unterschiedliche Bibliotheken, die oft ihre eigenen Datentypen mitbringen, einfach einbinden zu können, sind beliebige Datentypen erlaubt.

Zu jedem Ein- und Ausgang eines Glyphen wird der Datentyp festgelegt. Werden Glyphen mit unterschiedlichen Ein- und Ausgangsformaten miteinander verbunden, muß eine Formatkonvertierung vorgenommen werden. Prinzipiell wäre es möglich, spezielle Glyphen für die Konvertierung unterschiedlicher Formate zu erstellen. Der Nachteil davon wäre, daß der Benutzer sich um die Konvertierung kümmern muß und ein Teil der Flexibilität verloren geht. Um die Verwendung unterschiedlicher Datenformate

für den Benutzer transparent zu gestalten, werden die Konvertierungen vom Interpreter automatisch vorgenommen. Wird zum Beispiel ein Glyph mit einer Bilddatei (Typ file/tiff) als Ausgang, mit einem Glyphen verbunden, der als Eingang eine Referenz auf eine Image Klasse innerhalb des Python-Interpreters erwartet (Typ python/BWImage.Image), muß eine Formatkonvertierung von file/tiff nach python/BWImage.Image vorgenommen werden.

Der Interpreter hat im Modul Convert.py für jede mögliche Formatkonvertierung eine Funktion definiert. Um die Anzahl der möglichen Konvertierungen gering zu halten, wird für die quantitative Telepathologie file/tiff für den Datenaustausch mit der Datenbank verwendet, python/BWImage.Image für 2D Bildverarbeitung und python/vtkImageData für 3D Bildverarbeitung.

## Client

Für die Integration von Python-Skripten wurde ein neuer Typ von Glyph entwickelt, der Funktionalitäten zum Editieren von Skripten, für deren Übertragung zum Server und zur Ausführung der Skripte auf dem Server bereitstellt. Jeder Glyph hat ein Fenster, um die Parameter des Glyphen einzustellen; dieses wurde um einen Editor für Skripte erweitert (Abb 6.9). Mit dem Editor

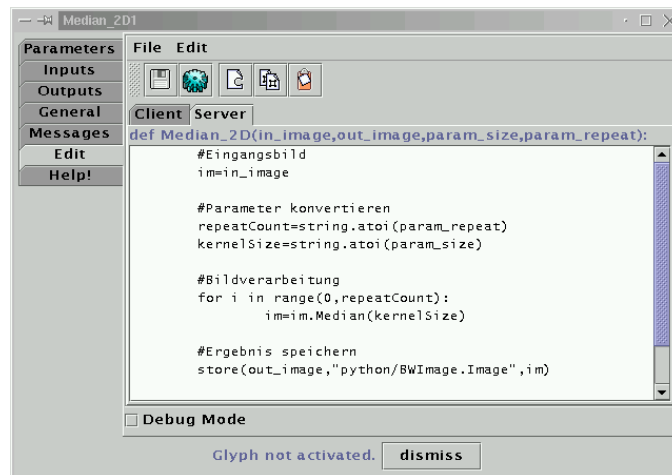


Abbildung 6.9: Editor zur Erstellung und Bearbeitung von Skripten.

lassen sich somit Skripte auf dem Client entwickeln und interaktiv testen. Durch die Verwendung einer Skriptsprache erübrigt sich bei der Erstellung



neuer Glyphen die Kompilierung. Glyphen lassen sich daher in erheblich kürzerer Zeit als z.B. mit Khoros entwickeln.

Der Editor in Abb. 6.9 zeigt beispielhaft ein Python-Skript für einen Median Glyphen. Der Glyph hat den Namen Median\_2D, in der Skriptdatei für den Glyphen Median\_2D.py ist eine Funktion

```
MEDIAN_2D(IN_IMAGE, OUT_IMAGE, PARAM_SIZE, PARAM_REPEAT)
```

definiert. Das Argument IN\_IMAGE enthält eine Referenz auf das Eingangsbild, OUT\_IMAGE ist ein Referenzname für das zu erstellende Ausgangsbild, PARAM\_SIZE enthält die Größe der Filtermaske aus dem Parameterfenster, und PARAM\_REPEAT gibt die Anzahl der Wiederholungen an. Der erste Schritt der Verarbeitung besteht im Einlesen der Daten. Dieser Schritt, inklusive eventueller Formatkonvertierungen wird vom Interpreter vorgenommen. Im zweiten Schritt werden die Parameter konvertiert. PARAM\_SIZE enthält einen String mit der Größe der Medianmaske, dieser wird durch `STRING.ATOI(PARAM_SIZE)` in einen Integer umgewandelt. Im dritten Schritt wird die Methode Median des Eingangsbildes aufgerufen, und im letzten Schritt wird mit

```
STORE(REFERENZNAME, TYP, REFERENZ)
```

eine Referenz auf das Ergebnis gespeichert.

## Server

Es wurde die 3-Tier Client/Server-Architektur, wie sie für Kommandozeilenprogramme verwendet wurde, beibehalten. Um ein Skript auszuführen, ruft der Glyph die entsprechende Methode der SD.Database auf. Die SD.Database sucht dann nach einem Server, um das Skript auszuführen und initiiert die Ausführung.

Um Python-Skripte auszuführen, wurde der Bildverarbeitungsserver erweitert. Für jeden angemeldeten Benutzer wird in einem Kindprozeß ein

Python-Skript gestartet, das mittels HTTP mit dem Java-Bildverarbeitungsserver kommuniziert. Mit HTTP-Anfragen werden Skripte aus einer Datei in den Server geladen oder Funktionen in dem Skript aufgerufen. Aus Sicherheitsgründen läuft der Bildverarbeitungsserver unter Linux als User nobody auf einem *seperaten* Server. Der Bildverarbeitungsserver ist durch eine Firewall so abgesichert, daß nur Verbindungen zum Server mit der SD.Database aufgebaut werden können.

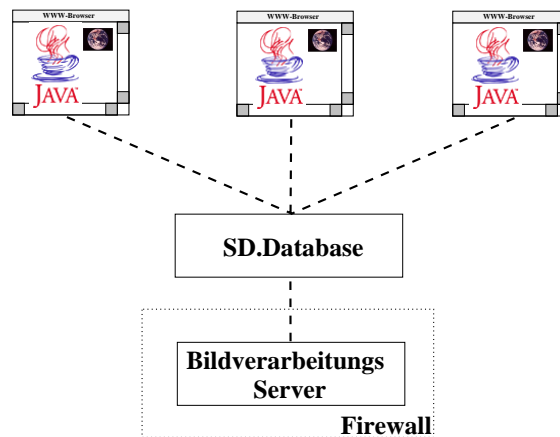


Abbildung 6.10: Der Bildverarbeitungsserver läuft auf einem separaten Rechner und ist durch eine Firewall abgesichert. Vom Bildverarbeitungsserver können nur Verbindungen zum Server mit der SD.Database aufgebaut werden.

## Erweiterungsmodule

Python kann durch *Extension modules* neue Funktionen integrieren, die sich dann, transparent für den Entwickler, in Skripten verwenden lassen. In der Praxis kommt es oft vor, daß Funktionen aus Bibliotheken z.B. zur Segmentierung in einem Skript für einen Glyphen benötigt werden. Python bringt eine eigene API mit, um solche Erweiterungsmodule zu erstellen. Dazu muß manuell ein Wrapper entwickelt werden.

Komfortabler, weil automatisch, gestaltet sich diese Arbeit aber mit dem *Simplified Wrapper Interface Generator* SWIG: Anhand eines ANSI-C Prototyps der zu integrierenden Funktionen erstellt SWIG den notwendigen Wrappercode, um die Funktion als Erweiterungsmodul in Python einzubinden.

Unterstützt wird die Einbindung von C, C++ und Objective-C Funktionen und Klassen in alle relevanten Skriptsprachen (Python, Perl, Tcl/Tk) [SWIG]. Die Vorteile von SWIG liegen auf der Hand:

**ANSI C/C++ Syntax:** Zur Generierung der Wrapper verwendet SWIG einen Prototypen der Funktion in ANSI C/C++. Dieser existiert bereits in der Headerdatei der Bibliothek und wird für SWIG übernommen. Nur für spezielle Fälle wie Compiler-Direktiven oder Templates müssen manuelle Modifikationen vorgenommen werden.

**Automatisierung:** SWIG generiert den kompletten Wrapper, inklusive der Anbindung an die Bibliotheksfunktionen. Es ist nicht notwendig, neben der Definitionsdatei Code manuell zu erstellen.

**C/C++ Optimierung:** SWIG ist auf die Einbindung von C/C++ Bibliotheken optimiert. Es ist daher selten notwendig, die Definitionsdatei zu modifizieren. Selbst eine große Bibliothek wie z.B. Open GL benötigt nur 3 Modifikationen und lässt sich in kürzester Zeit in Python integrieren (vergl. [SWIG Tutorial]).

Auf eine detaillierte Erläuterung zu SWIG wird an dieser Stelle verzichtet, sie findet sich im SWIG Dokumentation [SWIG]. Statt dessen wird anhand eines praktischen C Beispiels die Einfachheit von SWIG demonstriert:

SWIG Prototyp:	Headerfile der Bibliothek:
<pre>/* Name des Moduls */ %module example  /*Variablen*/ extern double My_variable;  /** Funktionsprototypen*/ extern long fact(long); extern int mod(int n, int m); extern char *get_time();</pre>	<pre>/* example.h*/  /*Variablen*/ extern double My_variable;  /** Funktionsprototypen*/ extern long fact(long); extern int mod(int n, int m); extern char *get_time();</pre>

Das Beispiel enthält einige einfache Funktionen, die in der Bibliothek `example.so` implementiert sind. Die SWIG-Prototypendatei enthält exakt die Headerdatei, ist aber um einige SWIG-spezifische Kommandos ergänzt (z.B. Name des Moduls). Nach der Übersetzung sind die Funktionen in Python verfügbar. Das Skript

```
from example import *
print get_time()
print cvar.My_variable
print fact(4)
```

liefert als Ergebnis:

```
Mar  1 10:31:42 2000
3.0
24
```

Analog zu diesem Beispiel werden auch größere C und C++ Bibliotheken in Python eingebunden.

### 6.5.2 Benutzeroberfläche

In der bisherigen Version des SD sind alle Parameter zu einem Glyphen statisch und werden vor der Ausführung des Glyphen in einem Fenster spezifiziert. Manche Glyphen erfordern aber eine erweiterte Benutzeroberfläche, um Parameter editieren zu können. Ein Beispiel hierzu ist die Veränderung der Grauwerte eines Bildes mit einer Lookup Tabelle (Abb. 6.11).

Eine Möglichkeit hierzu wäre die Entwicklung eines speziellen Glyphtyps für Lookup Tabellen, der entsprechende Eigenschaften mitbringt. Technisch stellt dies keine große Hürde dar, die `PyScriptGlyph` Klasse könnte abgeleitet werden, um sie dann um spezielle Eigenschaften zu erweitern. Solche Entwicklungen müßten aber auf dem Server vorgenommen werden und wären relativ aufwendig, da jeder Testlauf die Installation der Klassen auf dem Webserver erfordert.

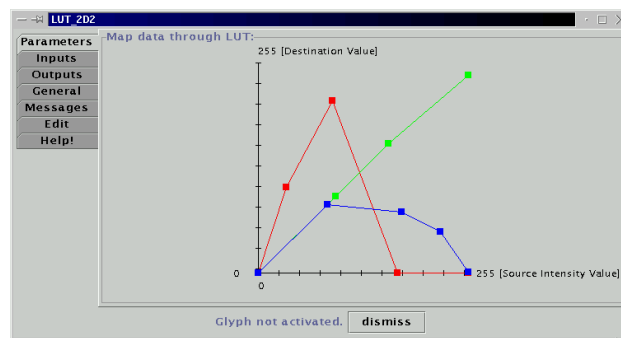


Abbildung 6.11: Benutzeroberfläche für den Glyphen *LUT\_2D* zur Veränderung der Grauwerte.

Erfreulicherweise ist Java inzwischen etwas gereift, und es existieren auch Skriptsprachen, deren Interpreter in Java geschrieben wurde.<sup>2</sup> Die Vorteile, die für den Einsatz von Skriptsprachen auf dem Server sprechen, gelten auch für den Client: Zu jedem Glyph läßt sich mit einem Skript eine individuelle Benutzeroberfläche erstellen. Ähnlich wie für den Server existiert auch ein Editor für clientseitig interpretierte Skripte. Skripte werden clientseitig interpretiert und realisieren Benutzeroberflächen durch die Verwendung von Java-Fenstern, Buttons etc. Die Skriptsprache erlaubt eine schnellere Entwicklung und ein sofortiges Testen ohne Neustart des SD.

Die Wahl von Python für serverseitige Skripte kam nicht von ungefähr, geschickterweise gibt es auch eine Implementierung von Python in Java, JPython genannt [JPython]. JPython läuft in derselben Java Virtual Machine wie der Client und hat transparenten Zugriff auf alle Javaklassen und Java-Objekte. Insbesondere können Pythonklassen von existierenden Javaklassen abgeleitet werden und Methoden von Javaklassen aufrufen. Für Benutzeroberflächen in (J)Python wird die Java-GUI-Bibliothek verwendet. Durch Vererbung von Javaklassen an Pythonklassen werden notwendige Eventhandler realisiert. Anbei ein kleines Beispiel für einen Taschenrechner mit JPython:

---

<sup>2</sup>Der Leser sollte sich nicht von JavaScript in die Irre führen lassen: JavaScript ist eine von Netscape entwickelte Skriptsprache. Sie kann nicht als Skriptsprache für Java verwendet werden.

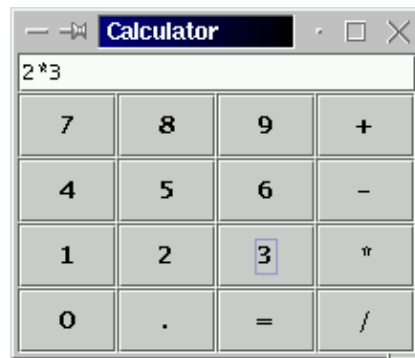


Abbildung 6.12: Mit JPython erstellter Taschenrechner.

```

def push(event): #Eventhandler für alle Buttons außer '='
    display.replaceSelection(event.actionCommand)
def enter(event): #Eventhandler für '=' Button
    #Berechnung d. Ergebnisses mit eval
    display.text=str(eval(display.text))
    display.selectAll()
#Panel mit Buttons erzeugen
keyPanel=JPanel(GridLayout(4,4))
for label in ['7','8','9','+', '4','5','6','- ',
'1','2','3','*', '0','.', '=', '/']:
    key=JButton(label)
    if label == '=':
        key.actionPerformed=enter #Eventhandler setzen
    else:
        key.actionPerformed=push#Eventhandler setzen
    keyPanel.add(key)
#Button zum Fenster hinzufügen, Fenster anzeigen
display=JTextField()#Textfeld für Display
panel= JPanel(BorderLayout())
panel.add("North",display)
panel.add("Center",keyPanel)
JFrame("Calculator",visible=1).getContentPane().add(panel)

```

Mit dem in Abbildung 6.9 gezeigten Editor werden Skripte auf dem Server und Skripte für die Glyph-spezifische Benutzeroberfläche erstellt. Für die

Implementierung einer Benutzeroberfläche leiten JPython-Skripte von der Javaklasse `SD.LANG.SDGLYPHGUIBASE` ab. Diese Klasse stellt zwei Möglichkeiten bereit:

Die Methode `getPane()` gibt eine `javax.swing.JPanel` Instanz zurück, die in den Reiter des Parameterfensters eingeordnet wird. Diese Methode lässt sich überschreiben, um ein eigenes Panel zurückzugeben. Ein Beispiel ist Parameterfenster (Abb. 6.13) des Glyphen `GetImage`, mit dem Bilder aus der Datenbank ausgelesen werden können.

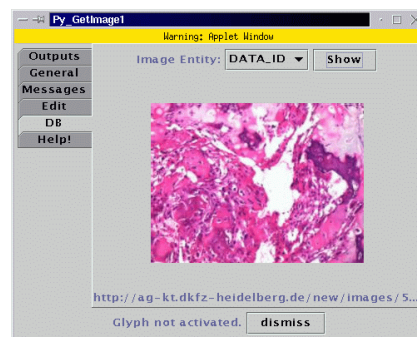


Abbildung 6.13: *Parameterfenster, um ein Bild aus der Datenbank auszuwählen. Aus den Bildentitäten (z.B. DATA\_ID) lässt sich ein Bild auswählen und anzeigen.*

Eine andere Möglichkeit ist die Übernahme der Kontrolle über die serverseitige Ausführung des Skriptes. Die Methoden `runBeforeServer`, `runServer` und `runAfterServer` werden aufgerufen, bevor der Server aktiviert wird, um den Server zu aktivieren bzw. nachdem der Server aktiviert wurde. Ein Beispiel ist der Glyph `PutImage` zum Ablegen von Ergebnisbildern in der Datenbank. Mit der Methode `runBeforeServer` legt der Client einen neuen Datensatz an, und die serverseitig ausgeführte Methode `runServer` überträgt die Daten an die Datenbank. Mit den hier vorgestellten Erweiterungen können innerhalb kurzer Zeit zu einem Glyphen individuelle Oberflächen erstellt werden.

Leider ist die clientseitige Implementierung von JPython als Interpreter in einem Java Applet nicht trivial: Der Interpreter versucht bei der Initia-

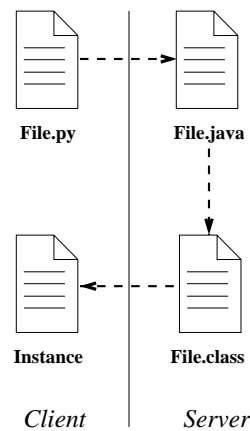


Abbildung 6.14: *Interpretation von JPython: Das JPython Skript File.py wird zum Server übertragen, in Java umgewandelt, kompiliert, als Bytecode vom Webserver zum Client übertragen und instanziiert.*

lisierung, alle Jar-Dateien<sup>3</sup>, die sich im Java-Classpath<sup>4</sup> befinden, zu lesen. Aufgrund der Einschränkungen für Applets bricht der Java Security Manager einen solchen Versuch mit einer SecurityException ab. Zwar ließen sich die erforderlichen Rechte einem Signed Applet einräumen, sind aber eigentlich unnötig. Es wird deshalb ein anderer Ansatz gewählt: Vor der Ausführung wird das JPython-Skript an den Server übertragen. Der Server erstellt aus dem JPython Quelltext Java Quelltext, kompiliert den Code und legt die daraus erstellten Klassen auf dem HTTP-Server ab. Vor der Instanzierung durch das Java Applet überträgt der Classloader die Klassen vom Webserver.

Der Python-Interpreter erstellt aus dem Python Sourcecode Java Bytecode, der von der JVM ausgeführt wird [Hungunin 97]. Für die Erstellung von Applets existiert aber die Möglichkeit, den Python Quelltext in Java Quelltext umzuwandeln, zu kompilieren und als Applet in Webseiten zu integrieren. Diese Besonderheit wird auf dem Server verwendet. Die Zeit zur Kompilierung des Python-Codes und zur clientseitigen Instanzierung liegt bei etwa 1 Sekunde (300 MHz Pentium II, Linux). Der kompilierte Python-

<sup>3</sup>Java Archiv (Jar) Dateien enthalten in einem komprimierten Archiv mehrere Java-Klassen.

<sup>4</sup>Der Java Classpath enthält Dateipfade, in denen der Java Classloader nach Klassen sucht. Der Classpath ist mit der Umgebungsvariablen LD\_LIBRARY\_PATH unter Unix vergleichbar.



Code ist für GUI Operationen nur unwesentlich langsamer als Java-Code. Die Performance von JPython ist von der verwendeten JVM abhängig. Bei Verwendung der Microsoft JVM 2.0 mit JIT-Compiler unter Microsoft Windows NT 4.0 ist JPython 1.7 mal langsamer als die C Implementierung von Python [Hungunin 97].

### 6.5.3 Report-Generierung für Glyphen

In der Praxis wird bei der Erstellung visueller Programme ein Großteil der Entwicklungszeit mit der Überprüfung der Ergebnisse einzelner Glyphen zugebracht, wobei meist immer nur die gleiche Art der Überprüfung vorgenommen wird. Unter Khoros wird dazu ein Programm zum Betrachten von Bildern mit einem Glyphen verbunden, um die Ergebnisse des Glyphen anzuzeigen. Der SD verfolgt einen praktischeren Ansatz: Im SD hat jeder Glyph einen *View*-Button, der auf dem Webserver eine Webseite mit Informationen zu diesem Glyph erzeugt und im Browser anzeigt. Diese Seite enthält neben den Bildern Daten über die Verarbeitung, z.B. den Threshold einer Segmentierung oder die Größe eines Bildes (Abb. 6.15).

Auf dem Server existiert zu jedem Glyph eine HTML-Seite mit *Server Side Include* (SSI) Befehlen. Die SSI Befehle enthalten Python-Code, der vom Server ausgeführt wird, bevor die Seite zum Client geschickt wird. Es wurde ein neuer HTML-Tag `<Py>` und `</Py>` definiert. Python-Code, der sich innerhalb dieser Tags befindet, wird vom Interpreter ausgeführt. Folgender HTML-Code

```
<H3>
Size:
<PY>
doc.write(str(inImage.x_Size())+"x"+str(inImage.y_Size()))
</PY>
</H3>
```

erzeugt z.B. eine Zeile mit der Größe des Bildes *inImage*:

Size: 768×576

Der HTTP-Server läuft im Python-Interpreter des Bildverarbeitungsservers. Innerhalb der HTML-Seite kann mit den SSI deshalb auf alle durch einen Glyph instanzierten Python-Objekte zugegriffen werden.

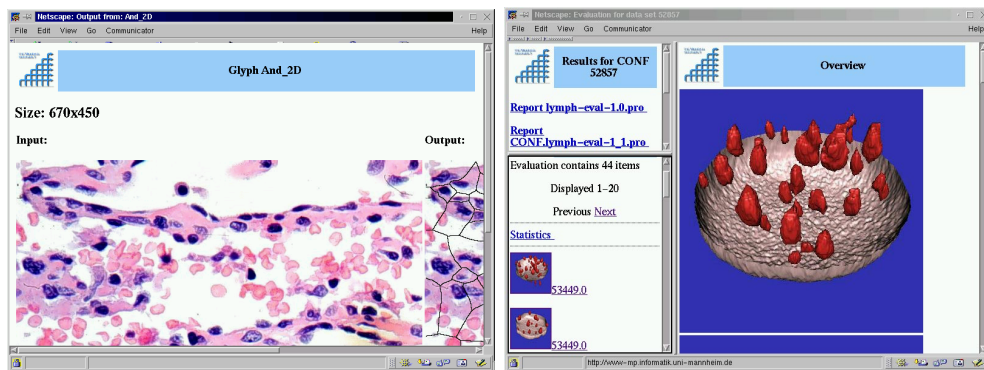


Abbildung 6.15: Links: Zur Betrachtung der Ergebnisse eines Glyphen wird eine HTML Seite mit Informationen zu den Ergebnissen erzeugt. Rechts: Report zu einer Analyse.

#### 6.5.4 Report-Generierung für Analysen

Bei der Anwendung eines Analyseverfahrens werden numerische Daten und Bilder von wesentlichen Zwischenschritten in der Datenbank gespeichert. Aufgabe der Report-Generierung ist die Darstellung der Zusammenhänge zwischen den Bilddaten und den quantitativen Daten. Dazu werden cgi-bin Python-Skripte eingesetzt, die auf dem Webserver dynamisch Seiten generieren. Im Browser werden drei Frames angezeigt (Abb. 6.15): Oben links sind alle zu einem Datensatz verfügbaren Analysen abrufbar, unten links sind Informationen zu den einzelnen analysierten Bildern abzurufen, die im dritten Frame rechts angezeigt werden.

Die dynamische Erzeugung der HTML-erfolgt analog zu der Report Generierung für Glyphen mit Python-Skripten, die Python SSI innerhalb von HTML-Seiten auswerten.

## 6.6 Anwendungen

Für die quantitative Telepathologie wurden einige Anwendungen für histologische 2D und 3D Bilder implementiert. Sie stützen sich auf Ergebnisse früherer, am DKFZ durchgeführter Diplom- und Doktorarbeiten. Auf eine Übersicht über die in QuanTed implementierte Glyphen wird an dieser Stelle verzichtet. Eine Übersicht über die wichtigsten Glyphen findet sich in Anhang B. Dort sind auch die verwendeten Segmentierungsverfahren beschrieben.

### 6.6.1 3D: Lymphozyten

Axel Bischoff hat in seiner Dissertation Verfahren zur Rekonstruktion und 3D-Darstellung der inneren Struktur von Zellkernen entwickelt [Bischoff 98]. Ziel dieser Entwicklung war die Messung der Verteilung von weißen Blutkörperchen (Lymphozyten), um zu untersuchen, ob Änderungen dieser Verteilung Hinweise auf das Vorhandensein von Krebs geben können. Die Lymphozyten sind Teil des menschlichen Immunsystems; es ist daher zu erwarten, daß sich Änderungen auf zellulärer Ebene als erstes auf die Lymphozyten auswirken.

Für die Aufnahme der Lymphozyten mit einem konfokalen Laserscanningmikroskop wurde Blutserum von gesunden und Krebspatienten verwendet. Bischoff konnte bei Krebspatienten weniger und kleinere Lymphozyten nachweisen als bei Kontrollpersonen. Details finden sich in [Bischoff 98].

Das visuelle Programm zur Analyse der 3D Daten ist in Abbildung 6.16 dargestellt. Die Bilddaten für die beiden Kanäle (Zellkern und Lymphozyten) werden aus dem Bildarchiv ausgelesen. Aus technischen Gründen sind die mit einem konfokalen Fluoreszenzmikroskop aufgenommenen Bilder verrauscht. Die Auswertung kleiner Strukturen wie z.B. der markierten Kinotechoren in Zellkernen wird dadurch erschwert. Um den Segmentierungsprozeß zu vereinfachen, werden die Daten mit einem von Donoho und Johnstone vorgeschlagenen Verfahren entrauscht [Donoho 92]. Dazu werden die Daten in eine Waveletdarstellung transformiert, Koeffizienten unterhalb

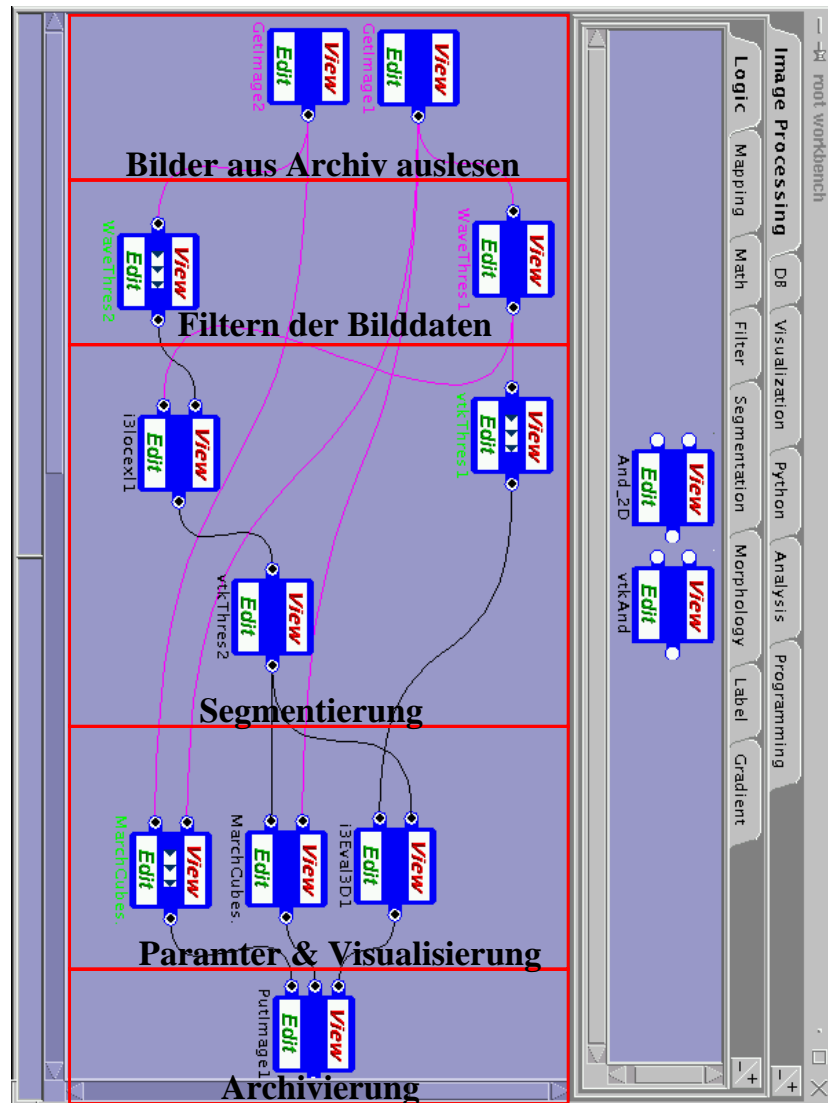


Abbildung 6.16: Programm zur Rekonstruktion und Visualisierung des Zellkerns und der Lymphozyten. Die Bilddaten werden mit den Glyphen `GetImage1` (Kanal 1) und `GetImage2` (Kanal 2) vom Server abgerufen und gefiltert (`WaveThres`). Die Segmentierung des Zellkerns (Kanal 1) erfolgt mit einem Threshold, die Segmentierung der Lymphozyten mit einem adaptiven Schwellwertverfahren. Der Glyph `i3Eval3D` ermittelt aus den segmentierten Bildern quantitative Daten zur Beschreibung der Lymphozyten. Zur Kontrolle werden zwei Visualisierungen der Daten durchgeführt und die Ergebnisse in der Datenbank archiviert.

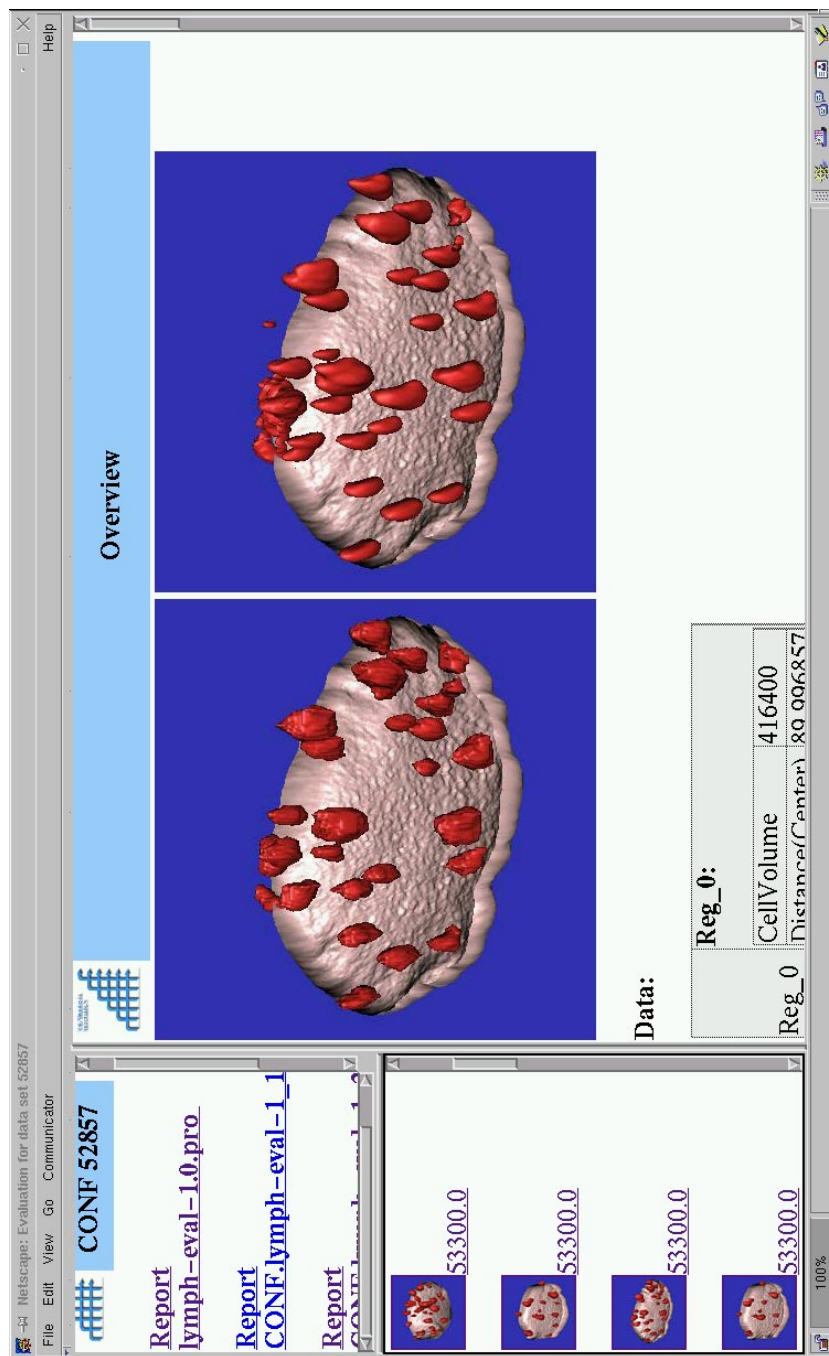


Abbildung 6.17: Report zur Darstellung der Ergebnisse des Analyseschritts. Zu jedem analysierten Bild werden die Ergebnisse der Segmentierung mit einem lokal adaptiven Schwellwert (linkes Bild) und zur visuellen Überprüfung der Segmentierung ein mit einem fixen Threshold segmentiertes Bild (rechtes Bild) dargestellt. Die Visualisierung erfolgt oberflächenbasiert mit dem Marching Cubes Verfahren.

eines Thresholds werden auf 0 gesetzt, und anschließend wird eine Rücktransformation durchgeführt. Die Segmentierung der Zellkerne (Kanal 1) erfolgt mittels eines Threshold-Verfahrens, die Segmentierung der Lymphozyten mit einem lokal adaptiven Threshold. Aus den beiden segmentierten Datensätzen berechnet der Glyph *i3Eval3D* Daten über Position und Verteilung der Lymphozyten. Zur Überprüfung der Segmentierungsergebnisse werden zwei Visualisierungen durchgeführt: Ein Bild zeigt die segmentierten Lymphozyten, das zweite Bild zeigt die Lymphozyten und den Zellkern. Im letzten Schritt werden die Ergebnisse in der Datenbank abgelegt und lassen sich als Report abrufen (Abb. 6.17).

### 6.6.2 3D: H19

Die Veränderungen von Genen können zu einer Fehlsteuerung des Zellwachstums führen. Das Verständnis der Zusammenhänge zwischen Genveränderungen und Funktionsänderungen ist Gegenstand aktueller Untersuchungen in der Krebsforschung. Für die Diagnostik wird das Wissen über Genveränderungen als Tumormarker verwendet. Ein Beispiel hierfür ist das Gen H19. Die Aktivität dieses Gens korreliert mit dem Malignitätsgrads von Harnblasentumoren [Fischer 96].

Die genaue Funktion des Gens ist noch nicht geklärt [Ariel 95]. Es wird besonders in der embryonalen Entwicklung exprimiert, postnatal ist es aber nicht mehr aktiv. Mit einem konfokalen Laserscanning Mikroskop wurden 3D-Aufnahmen von Zellkernen gemacht, bei denen der Zellkern mit Probidiumjodid eingefärbt und das Gen H19 mit Cy 2 markiert war. Ziel war die Messung der Aktivität von H19 in Zellkernen. Das visuelle Programm zur Auswertung ist analog zu dem in Abb. 6.16.

### 6.6.3 2D: Kerngrading

Die Bildverarbeitung analysiert histologische Bilder und ermittelt Parameter zur Beschreibung von Zellkernen. Dies erfolgt entweder auf der Ebene von einzelnen Zellkernen, oder topologisch auf der Ebene des Gewebes zur

Beschreibung mehrerer Zellkerne im Objektverbund.

Im ersten Fall werden von stark vergrößerten Zellkernen (Objektiv  $100\times$ ) die morphologischen Merkmale wie die Anzahl der Epithelobjekte, Fläche, Anzahl, Form und Konturmaße der Drüsenlichtungen ermittelt; im zweiten Fall werden die in Anhang B beschriebenen Merkmale bestimmt d.h. es werden Parameter zur Beschreibung der Zellkerne im Objektverbund wie Voronoi-Partitionen, Ellipsenanpassungen und Polarität ermittelt.

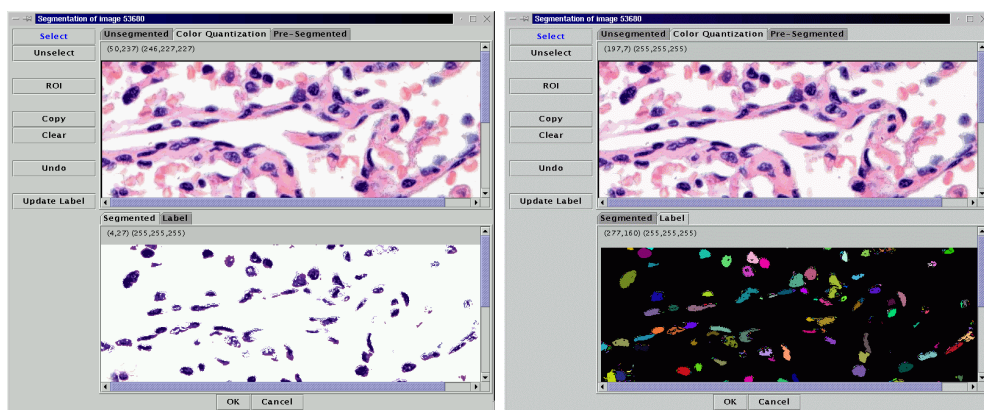


Abbildung 6.18: *Tool zur Grobsegmentierung histologischer Bilder: Auf dem Server werden von einem histologischen Bild zwei Farbsegmentierungen durchgeführt. Im oberen Teil des Fensters werden das ursprüngliche Bild, ein übersegmentiertes und ein untersegmentiertes Bild angezeigt. Der Benutzer kann durch mit der Maus Bereiche und Farben zum segmentierten Bild, das im unteren Teil des Fensters angezeigt wird, hinzufügen (links). Zur Kontrolle wird ein Labelbild angezeigt, auf dem sich überlappende Zellkerne leicht erkennen lassen (rechts). Überlappender Zellkerne können im Labelbild mit der Maus getrennt werden.*

Abbildung 6.19 zeigt das visuelle Programm zur Bestimmung topologischer Merkmale, wie sie in [Fischer 96] beschrieben sind. Als Bilddaten werden farbige histologische Bilder verwendet, die vom Benutzer mit einem Segmentierungstool vorsegmentiert wurden (Abb. 6.18). Die Bilddaten für das histologische Bild werden aus der Datenbank abgerufen (GetImage2, Bild 1), dazu das vorsegmentierte Bild (GetImage1). Vom vorsegmentierten Bild werden alle Regionen mit der Größe unterhalb eines Thresholds herausgefiltert (Bild 2, Bild 3), um anschließend die Voronoi-Partitionen zu bestimmen und

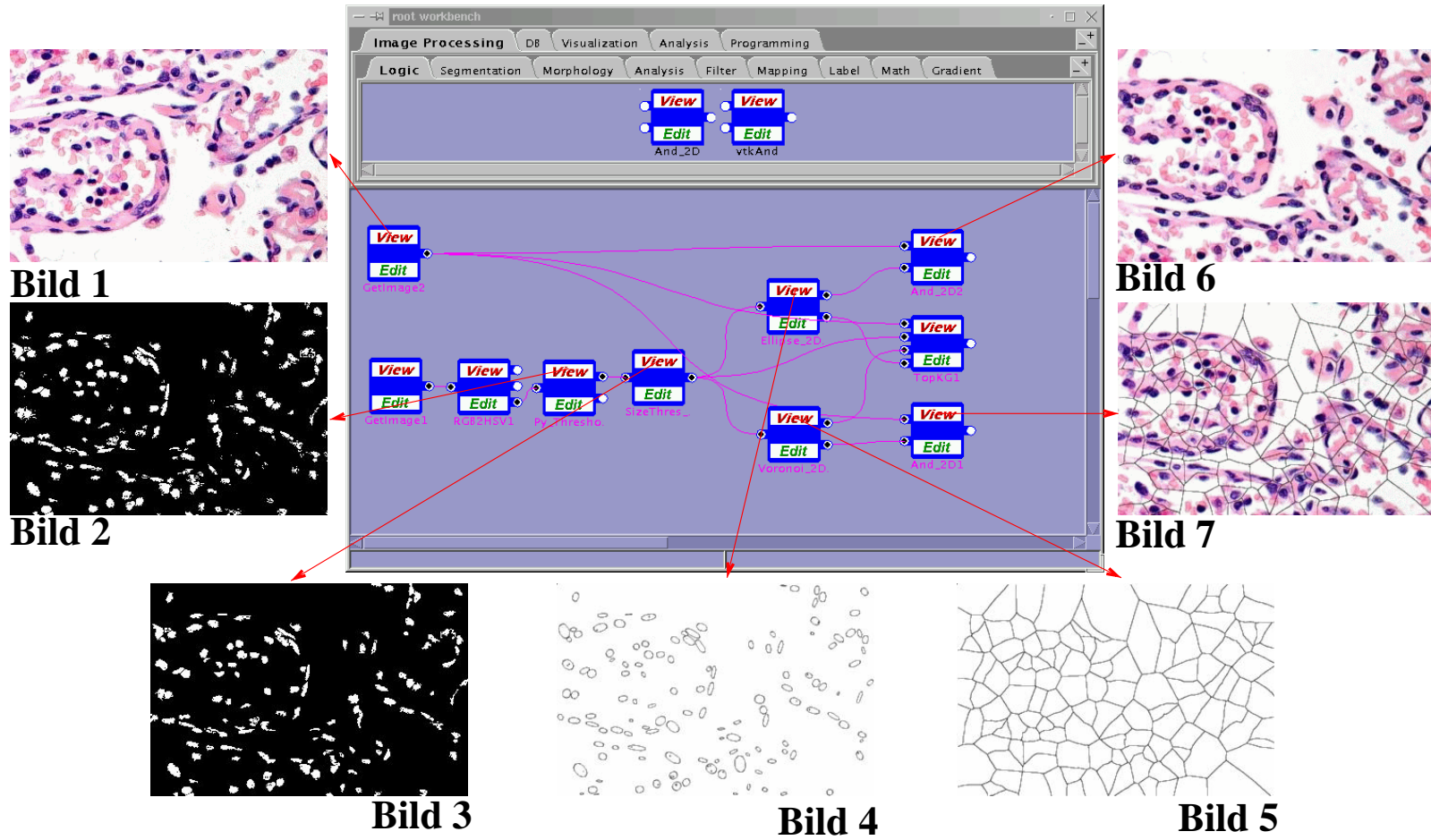


Abbildung 6.19: Visuelles Programm zum topologischen Kerngrading.



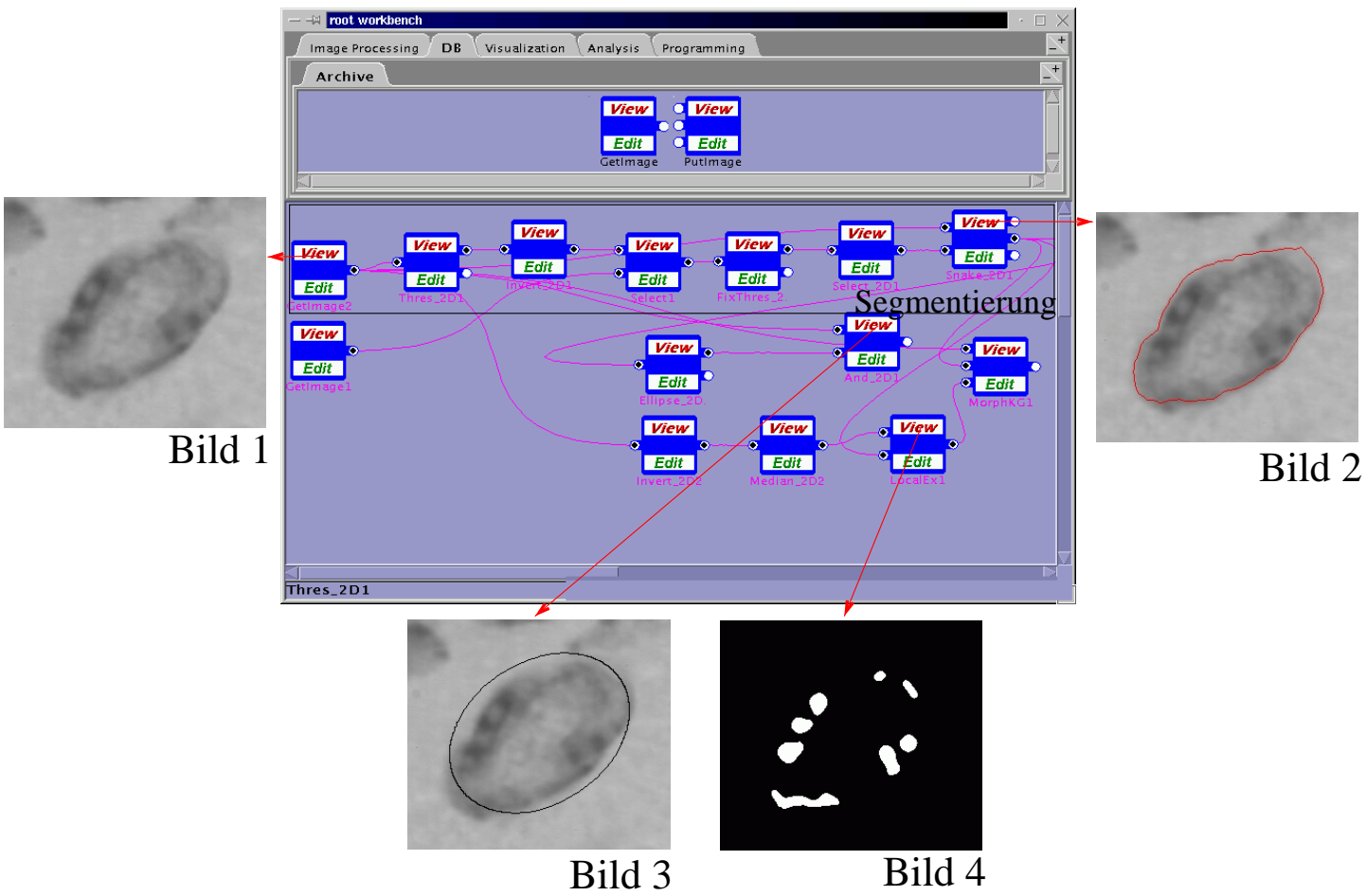


Abbildung 6.20: Visuelles Programm zum morphologischen Kerngrading.

eine Ellipsenanpassung vorzunehmen (Bild 4, Bild 5). Die Ellipsenanpassung und die Voronoi-Partitionen werden in das histologische Bild eingezeichnet (Bild 6, Bild 7). Die Parameter aus der Ellipsenanpassung und den Voronoi-Partitionen werden mit dem Glyph TopKG berechnet. Als Ergebnis werden Bild 6 und Bild 7 zusammen mit den durch den Glyphen TopKG ermittelten Parametern in der Datenbank gespeichert. Die Ergebnisse können als Report abgerufen werden (Abb. 6.21).

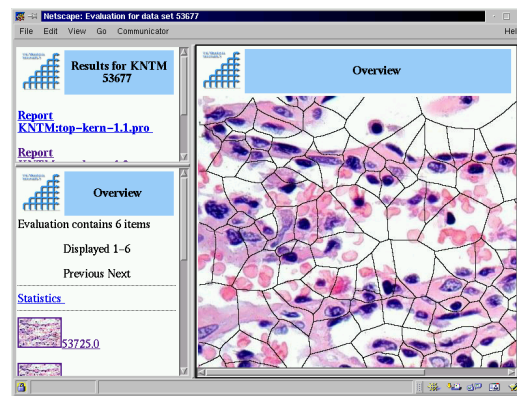


Abbildung 6.21: *Report zum topologischen Kerngrading.*

Das visuelle Programm zur Bestimmung morphologischer Merkmale ist in Abb. 6.20 gezeigt. Als Bilddaten werden Grauwertbilder von stark vergrößerten Zellkernen verwendet. Diese wurden in den vergangenen Jahren von MTAs am DKFZ aufgenommen und z.T. manuell segmentiert.

Die Bilddaten für den Kern werden aus der Datenbank ausgelesen (Bild 1). Falls ein vorsegmentiertes Bild vorhanden ist, wird dieses für die weitere Segmentierung verwendet. Andernfalls wird mit einem Thresholdverfahren eine Grobsegmentierung vorgenommen. Die Feinsegmentierung wird mit aktiven Konturmodellen (Snakes) durchgeführt (Bild 2). Anschließend wird eine Ellipsenanpassung vorgenommen (Bild 3) und mit einem Lokalen Extremwertfinder die Chromatinregionen bestimmt (Bild 4). Die Einzelheiten des Verfahrens sind in [Janson 96] beschrieben.

## 6.7 Klassifikation

Während die einzelnen Bildverarbeitungsschritte wie Filterung, Segmentierung und Parameterextraktion schon sehr ausgereift sind, fehlt am Ende noch ein Modul zur Interpretation der Ergebnisse. Aus mathematischer Sicht handelt es sich dabei um ein Klassifizierungsproblem: die gemessenen Daten sollen einer Klasse, d.h. einem Malignitätsgrad, zugewiesen werden. Gesucht wird also ein Klassifikator, der anhand vorher manuell klassifizierter Daten neue Fälle möglichst richtig klassifiziert.

Hierzu gibt es unterschiedliche Ansätze. [Fischer 96] verwendet einen Nächste-Nachbarn Ansatz, um aus den quantitativen Daten Rückschlüsse auf den Malignitätsgrad zu ziehen. Von einem Pathologen wurde der Malignitätsgrad von Gewebeproben unterschiedlicher Malignität bestimmt. Für diese Gewebeproben wurden mit der Bildverarbeitung quantitative Parameter extrahiert. Für die Klassifizierung neuer Fälle werden die quantitativen Daten dieser Fälle mit den manuell klassifizierten Daten verglichen. Bei seiner Studie mit 22 Proben konnte Fischer alle Proben exakt klassifizieren. [Janson 96] und [Stotzka 96] haben Neuronale Netze in Kombination mit statistischen Merkmalen für die Klassifizierung verwendet. Bei einer ausreichend großen Datenmenge zum Training der Netze kommen sie zu einer korrekten Erkennungsrate von etwa 80%.

Bisher wurden die Ergebnisse der quantitativen Parameter nur an kleinen Studien validiert; Fischer verwendete 22 Proben, Bischoff Lymphozyten von 6 Patienten und 2 Kontrollpersonen. Um eine fundierte, allgemeine Aussage treffen zu können, ist es notwendig die Verfahren in einer klinischen Studie an großen Datenmengen zu überprüfen und geeignete Klassifikatoren zu verwenden.

## 6.8 Zusammenfassung

In Kapitel 6 wurde QuanTed, ein E-Service für quantitative Telepathologie, vorgestellt. Es ist das erste System für quantitative Telepathologie überhaupt. Anstelle von Person-to-Person Konsultationen zwischen Konsultant

und Referent werden Services angeboten, die ein Referent ohne Unterstützung durch einen Konsultanten nutzen kann. Im Falle der quantitativen Telediagnostik ist der Service die Ermittlung von Parametern zur quantitativen Beschreibung von Gewebestrukturen. Dazu werden wie bei der statischen Telediagnostik Bilddaten auf dem Server abgelegt, und der Benutzer wählt ein Verfahren zur Analyse der Daten aus. Die Daten werden auf dem Server analysiert und die Ergebnisse in der Datenbank gespeichert. Die Analyseergebnisse werden in einem Report zusammengestellt und informieren über quantitative Parameter und Zwischenergebnisse der Bildverarbeitung.

Das in den vorigen Kapiteln vorgestellte Framework wurde um eine Entwicklungsumgebung zur interaktiven Bildverarbeitung erweitert. Über das Web lassen sich innerhalb dieser Entwicklungsumgebung neue Verfahren zur Extraktion quantitativer Parameter aus den Bilddaten entwerfen, um sie dann später auf andere Datensätze anzuwenden. Für die Bildverarbeitung wurde ein visueller Programmieransatz gewählt. Jede Bildverarbeitungsoperation wird durch ein grafisches Symbol, einen Glyphen, dargestellt. Die Glyphen werden untereinander verbunden und stellen den Datenfluß zwischen den einzelnen Operationen dar. Aufgrund der visuellen Programmierung lassen sich sehr schnell neue Verfahren entwerfen, da sich der Entwickler statt auf die Programmierung auf die Bildverarbeitung konzentrieren kann. Die visuelle Programmierung fördert das Ausprobieren unterschiedlicher Operationen und gibt einen schneller Überblick über die Ergebnisse.

Es wurden unterschiedliche kommerzielle Systeme für visuelle Programmierung auf ihre Eignung für den Einsatz in der webbasierten quantitativen Telepathologie überprüft. Obwohl sie prinzipiell gut für die Bildverarbeitung geeignet sind, sind sie aufgrund der speziellen Anforderungen des Webs für die quantitative Telepathologie ungeeignet. Es wurde deshalb speziell für den Einsatz im Web eine visuelle Programmierumgebung entwickelt. Sie zeichnet sich insbesondere durch ihre Flexibilität aus. Anstelle einer starren API zur Erweiterung des Systems durch neue Glyphen wird die Skriptsprache Python eingesetzt. Im Vergleich zu Entwicklungen neuer Glyphen mit C/C++ lassen sich Glyphen durch die Verwendung einer Skriptsprache sehr viel schneller und interaktiver testen und entwickeln, als es mit kommerziel-

len Systemen und C/C++ möglich ist. Bildverarbeitungsoperationen werden in C/C++ implementiert und als Erweiterungsmodule in Python integriert. Dieser Schritt wird durch den Wrappergenerator SWIG automatisiert. Das typische Skript für einen Glyphen besteht aus etwa 5-50 Zeilen und enthält 1-10 Aufrufe an eine Bildverarbeitungsbibliothek.



# Kapitel 7

## Diskussion und Ausblick

Gegenstand dieser Arbeit war der Entwurf und die Implementierung eines Telemedizinssystems für die Telediagnostik in der Pathologie. Die Telemedizin befaßt sich mit der Erbringung medizinischer Dienste zwischen Arzt/Patient oder zwischen Ärzten, bei denen Konsultant und Patient räumlich getrennt sind. Sie läßt sich in fünf Bereiche unterteilen: Telediagnostik, Teletherapie, Telemonitoring, Teleeducation und Teleadministration. Das im Rahmen dieser Arbeit entstandene System WebTed (Webbasierte Telediagnostik) fällt in den Bereich der Telediagnostik.

Die Pathologie beschäftigt sich mit der naturwissenschaftlichen Beschreibung der Ursachen, Entstehungsmechanismen und morphologischen Manifestationen von Krankheiten. Dabei stützt sich die Diagnostik in der Pathologie hauptsächlich auf die Untersuchung von Gewebeproben bezüglich Abweichungen der Zellstruktur von der Norm. Die Untersuchungen werden von einem Pathologen unter einem Mikroskop vorgenommen.

Das WebTed-System implementiert drei unterschiedliche Ansätze für die Telediagnostik: die statische, die dynamische und die quantitative Telediagnostik. Statische Telediagnostik wird für Expertenkonsultationen eingesetzt. Mit einem Webbrowser werden klinische Daten und Bilder von Ausschnitten eines histologischen Präparats von einem Referenten zu einem Fall zusammengestellt und auf dem WebTed-Server abgelegt. Mögliche Konsultanten werden per E-Mail benachrichtigt und erstellen eine Diagnose oder

beraten den Fall gemeinsam in einer Konferenzsitzung.

Nachteil der statischen Telediagnostik ist die feste Vorgabe der durch den Referenten selektierten Ausschnitte eines Präparats. Diesen Nachteil behebt die dynamische Telediagnostik. Der Konsultant bedient ein ferngesteuertes Mikroskop und hat selber die Kontrolle über die Bildselektion. Dabei lassen sich alle wesentlichen Funktionen des Mikroskops wie z.B. Fokus, Objektiv und Licht ferngesteuert bedienen.

Bei der statischen und dynamischen Telediagnostik basiert die Diagnose hauptsächlich auf der visuellen Untersuchung von Gewebeproben unter einem konventionellen Lichtmikroskop. Dazu wird eine Gewebeprobe durch einen Pathologen auf Merkmale wie einer veränderten Zellorganisation oder durch spezielle Marker hervorgerufene Färbungen hin untersucht. Ein Problem stellt die Fülle der Informationen eines histologischen Bildes dar. Bei der Analyse histologischer Bilder kommt es zu einer Abstrahierung, die zu einer subjektiven Selektion und Zuordnung von Merkmalen führt. Die Subjektivität kann jedoch durch Methoden der computergestützten Morphometrie reduziert werden: Mit Methoden der digitalen Bildverarbeitung werden Parameter zur Charakterisierung der Gewebestrukturen ermittelt. Diese Parameter sind objektiv und jederzeit reproduzierbar und tragen deshalb zu einer Objektivierung der Diagnose bei.

Kein anderes der z.Zt. verfügbaren Telediagnostiksysteme unterstützt die Extraktion quantitativer Parameter aus histologischen Bildern mit Methoden der digitalen Bildverarbeitung. Als erstes System integriert WebTed sowohl statische und dynamische, als auch quantitative Telepathologie. In den folgenden Abschnitten werden die einzelnen Komponenten für statische, dynamische und quantitative Telepathologie kritisch betrachtet und mögliche Verbesserungen und Alternativen aufgezeigt.

## Architektur

Das WebTed-System verwendet eine Client/Server-Architektur. Mit einem Webbrowser wird ein Java Applet vom WebTed-Server abgerufen. Das Applet nutzt für die Realisierung von statischer, dynamischer und quantitativer



Telepathologie verschiedene Serverdienste wie z.B. Datenbankdienste, Steuerung unterschiedlicher Mikroskopfunktionen und Bildverarbeitungsdienste. Für die Implementierung der Client/Server-Architektur werden Java Remote Method Invocation (RMI) und CORBA verwendet.

Insgesamt hat sich die gewählte Architektur und der webbasierte Ansatz mit Java als für die Problemstellung gut geeignet erwiesen. Durch den webbasierten Ansatz werden Probleme bisheriger Systeme vermieden, bei denen die Wahl des Konsultationspartners auf diejenigen Partner eingeschränkt ist, die die gleiche Software benutzen. Zur Nutzung des WebTed Systems wird außer einem Webbrowser keine weitere Software benötigt.

## **Statische Telediagnostik**

Die statische Telediagnostik wird durch die Komponente StaTed (Statische Telediagnostik) realisiert. Mit einem Java Applet lassen sich von einem Referenten neue Fälle anlegen und verwalten. Bilddaten können entweder direkt von einer am Rechner angeschlossenen Videokamera, oder als Datei zu einem Fall hinzugefügt werden. Das Java Applet legt klinische Daten und Bilddaten auf den WebTed-Server ab. Konsultanten rufen den Fall mit ihrem Webbrowser von dort ab und können ihre Diagnose direkt über das Web eingeben.

Alternativ läßt sich ein Fall auch in einer Konsultationssitzung von mehreren Spezialisten gemeinsam in einer Online-Konferenz besprechen. Dazu wird jedem Teilnehmer einer Konferenz ein Fenster angezeigt, in dem ein Bild angezeigt wird. Während der Diskussion kann mit Pfeilen, die bei allen Mitgliedern angezeigt werden, auf Besonderheiten im Bild hingewiesen werden.

Die statische Telediagnostik erfüllt die Anforderungen des Bundesdatenschutzgesetzes. Alle zwischen Client und Server übertragenen Daten werden durch Verwendung von Secure Socket Layers (SSL) verschlüsselt übertragen. Klinische Daten werden auf dem Server in einer Adabas SQL-Datenbank gespeichert. Es werden dabei die in die SQL-Datenbank integrierten Sicherheitsmechanismen verwendet: Benutzer erhalten durch Verwendung von

Sichten (SQL Views) nur Zugriff auf diejenigen Daten, die für sie durch einen Referenten freigegeben wurden. Obwohl dadurch bereits ein sehr hoher Sicherheitsstandard erreicht ist, kann durch Angriffe auf den Server nicht ausgeschlossen werden, daß Unbefugte Zugriff auf die Daten erhalten.

Einen sehr viel höheren Sicherheitstandard ließe sich durch die Verwendung von asymmetrischen Verschlüsselungsverfahren und durch die verschlüsselte Speicherung aller Daten in der Datenbank erreichen. Public-Key-Verfahren verwenden zwei Schlüssel, einen privaten und einen öffentlichen. Nachrichten, die mit dem öffentlichen Schlüssel verschlüsselt wurden, können nur mit dem privaten Schlüssel entschlüsselt werden. Die Sicherheit der Public-Key Kryptographie beruht darauf, daß es nur mit großem Aufwand möglich ist, den einen Schlüssel aus dem anderen zu berechnen.

Alle Daten zu einem Fall ließen sich mit dem öffentlichen Schlüssel vom Referenten verschlüsseln und in der Datenbank ablegen. Der Konsultant kann mit seinem privaten Schlüssel die Daten wieder entschlüsseln (siehe [Rawolle 99]). Das Problem stellte die Schlüsselverwaltung dar: Um einen Fall mehreren Konsultanten zur Verfügung zu stellen, müssen die Konsultanten sich entweder auf einen gemeinsamen Schlüssel einigen oder die Daten mehrfach in der Datenbank ablegen und für jeden Konsultanten separat verschlüsseln. In der Praxis stellen sich dabei folgende Probleme: Es müßte eine Zertifizierungsinstanz zur Schlüsselverwaltung betrieben werden, die allerhöchste Sorgfalt bei der Identifikation der Clients und der Geheimhaltung der Zertifikate walten läßt. Zur eindeutigen Identifikation eines Clients bei der Zertifizierungsinstanz ist die Verwendung von Zertifikaten notwendig. Diese müßten lokal gespeichert werden, was mit Java Applets nur möglich ist, wenn dem Applet besondere Privilegien eingeräumt werden (Signed Applet)<sup>1</sup>.

Leider war weder der Aufbau einer solchen Identifizierungsinstanz mit den verfügbaren Ressourcen möglich, noch war es möglich auf eine kommerzielle Zertifizierungsinstanz wie z.B. Verisign oder Trustcenter zurückzugreifen.

---

<sup>1</sup>Zwar verfügen sowohl Netscape Communicator als auch Microsoft Internet Explorer über Mechanismen zur lokalen Schlüsselverwaltung, diese lassen sich aber nicht verwenden, da die verwendete Java Virtual Machine des Java Plugins von Sun keinen Zugriff auf die lokalen Zertifikate erlaubt.

Um dennoch eine hohe Sicherheit zu erreichen, werden alle Daten per SSL verschlüsselt übertragen und nur die unbedingt notwendigen Dienste sind auf dem Server verfügbar (siehe Anhang A).

### **Dynamische Telediagnostik**

Die dynamische Telediagnostik wird durch die Komponente DynTed (Dynamische Telediagnostik) bereitgestellt. Über ein Java Applet lassen sich alle wesentlichen Funktionen eines Mikroskops (Position, Fokus, Objektiv, Licht) über das Web fernsteuern. Am Mikroskop ist eine Farbvideokamera angeschlossen. Für die Übertragung und Anzeige der Videobilder stehen zwei unterschiedliche Aufnahmemodi zur Verfügung: ein dynamischer Modus und ein semi-dynamischer Modus. Beim semi-dynamischen Modus werden Einzelbilder aufgenommen und im Client zu einem Gesamtbild zusammengesetzt. Es können dadurch einerseits Gesamtbilder beliebiger Größe aus Einzelbildern zusammengesetzt werden, andererseits lassen sich durch Caching Bilder wiederverwenden und die Bilddaten effizienter nutzen.

Der dynamische Modus überträgt ein Graustufenbild der Videokamera. Das übertragene Bild hat  $1/4$  der Größe eines normalen Videobildes ( $768 \times 576$ ). Der dynamische Modus wird zur Fokussierung und zum Suchen eines Ausschnitts verwendet, um dann mit dem semi-dynamischen Modus das Bild in voller Auflösung zu betrachten.

Im Vergleich zu anderen webbasierten Lösungen unterstützt DynTed als erstes System sowohl einen dynamischen als auch einen semi-dynamischen Modus und kann mehrere Einzelbilder zu einem bildschirmfüllenden Gesamtbild zusammensetzen. Vergleichbare Systeme besitzen nur einen semi-dynamischen Modus, bei dem jeweils Einzelbilder aufgenommen und übertragen werden.

Ein prinzipielles Problem ist die schnelle Darstellung von Videosequenzen in Java. Sun stellt hierfür das Java Media Framework (JMF) bereit, das leider zum Zeitpunkt der Implementierung noch nicht verfügbar war. Für die Videoübertragung werden deshalb animierte Gif-Bilder übertragen, die in einem Fenster des Browsers angezeigt werden. Im lokalen Netzwerk wird

damit eine Wiederholrate von etwa 7 Hz erreicht. In einer zukünftigen Erweiterung wäre es ratsam, das JMF zu verwenden. Mit dem JMF lassen sich Videobilder mit dem MPEG-Verfahren komprimieren und erlauben eine effizientere Datenübertragung und die Übertragung von Farbbildern.<sup>2</sup>

Eine sinnvolle Erweiterung der dynamischen Telediagnostik wäre die Integration eines Bildverarbeitungsfilters zur Verbesserung der Tiefenschärfe: Die für eine Diagnose relevanten Zellkerne befinden sich in unterschiedlichen Tiefen eines Präparats. Für die Befundung eines Präparats ist es deshalb erforderlich, Bilder des gleichen Ausschnitts aber unterschiedlicher Fokusebenen zu übertragen. Da das verwendete Mikroskop einen motorisierten z-Antrieb besitzt, wäre es möglich Bilder des gleichen Ausschnitts bei unterschiedlichen z-Positionen aufzunehmen und zu einem Extended-Focus Bild zu kombinieren, in dem alle Bereiche fokussiert sind. Verfahren dazu sind z.B. von [Tympel 95] vorgeschlagen worden.

Durch die Verwendung eines solchen Verfahrens ließe sich die mehrfache Übertragung des gleichen Ausschnitts mit unterschiedlichen z-Positionen vermeiden. Bisher ist ein derartiger Ansatz noch in keinem der verfügbaren Systeme integriert.

### **Quantitative Telediagnostik**

Die Quantitative Telediagnostik wird durch die QuanTed (Quantitative Telediagnostik)-Komponente implementiert. Die QuanTed-Komponente verfolgt einen anderen Ansatz als die statische und quantitative Telepathologie: Anstelle einer Konsultation zwischen Referent und Konsultant, d.h. einer Person-to-Person Konsultation, treten elektronische Dienste, die ein Referent ohne Unterstützung eines Konsultanten nutzen kann. Beim QuanTed-System ist dieser Service die Extraktion quantitativer Parameter zur Charakterisierung von Gewebestrukturen aus histologischen Bilddaten.

Für die Analyse der Daten stehen zwei Möglichkeiten zur Verfügung. Mit einer integrierten visuellen Programmierungsumgebung können Bildverarbei-

---

<sup>2</sup>Die vom Gif-Format unterstützten 256 Farben sind unzureichend für eine gute Darstellung von histologischen Bildern. Die verwendete Graustufendarstellung liefert bessere Kontraste und ist deshalb für Ausschnittsselektion und Fokussierung besser geeignet.

tungsspezialisten aus existierenden Bildverarbeitungsmodulen (z.B. Bildfilterung, Bildsegmentierung etc.) neue Analyseverfahren erstellen und diese in einer Methodendatenbank speichern. Die Anwendung der Verfahren erfolgt durch einen Pathologen. Zur Ermittlung quantitativer Parameter werden Bilddaten auf dem WebTed-Server abgelegt und der Benutzer wählt ein Verfahren zur Analyse der Daten aus. Die Daten werden auf dem Server analysiert und die Ergebnisse in der Datenbank gespeichert. Die Ergebnisse der Analyse werden in einem Report zusammengestellt. Sie informieren über quantitative Parameter und Zwischenergebnisse der Bildverarbeitung.

Für die Bildverarbeitung wurde ein visueller Programmieransatz gewählt, bei der jede Bildverarbeitungsoperation durch ein graphisches Symbol, einen Glyphen, dargestellt wird. Die Glyphen werden untereinander verbunden, um den Datenfluß zwischen den Bildverarbeitungsoperationen zu repräsentieren. Da kommerziell verfügbare Lösungen für die visuelle Programmierung aufgrund der speziellen Sicherheitsansprüche für die Verwendung im Web ungeeignet waren, wurde eine auf Java basierte visuelle Programmierumgebung entwickelt. Diese verwendet eine Client/Server-Architektur, bei der die Bildverarbeitungsoperationen auf dem Server berechnet werden und der Client zur Steuerung und zur Darstellung der Ergebnisse verwendet wird. Zur einfachen und flexiblen Erweiterung der Funktionalität werden für die Implementierung der Glyphen sowohl für den Client, als auch für den Server die Skriptsprache Python verwendet. Um dennoch Bildverarbeitungsoperationen schnell auszuführen, werden diese in C/C++ implementiert und als Erweiterungsmodule in Python integriert. Dadurch läßt sich die Funktionalität durch Hinzufügen neuer Glyphen einfach erweitern.

In verschiedenen Doktorarbeiten am DKFZ wurden Studien durchgeführt, die belegen, daß die mit Methoden der digitalen Bildverarbeitung ermittelten quantitativen Parameter zur Beschreibung der Zellkerne und Gewebeorganisation geeignet sind, ein computergestütztes histologisches Grading durchzuführen. Beispiele sind die Arbeit von M. Fischer oder C. Janson [Fischer 96] [Janson 96]. Bisher haben diese Methoden noch nicht den Weg in die Praxis gefunden. Hierfür gibt es zwei Gründe: Einerseits wurden die Ergebnisse nur in kleinen Studien verifiziert, so daß es möglich war die Bildverarbeitung

auf die vorhandenen Proben zu optimieren. Für eine allgemeine Verwendung sind deshalb an großen Datenmengen erprobte Verfahren notwendig. Andererseits wurden die Verfahren durch Unix Shellskripte implementiert und erforderten teilweise ein manuelles Eingreifen, so daß ihre Anwendung nur Spezialisten verständlich war.

Mit der in dieser Arbeit entwickelten Komponente QuanTed zur quantitativen Telediagnostik stehen die Werkzeuge zur Verfügung, die an kleinen Studien evaluierten Untersuchungen in großen Studien verifizieren und allgemein verfügbar machen zu können. Dazu bieten sich mit WebTed gute Möglichkeiten: Über das Web läßt sich eine große Anzahl verschiedener Fälle sammeln und von verschiedenen Pathologen der Malignitätsgrad bestimmen. Mit den ermittelten quantitativen Parametern und den von verschiedenen Pathologen bestimmten Malignitätsgrad der Präparate kann ein geeigneter Klassifikator entwickelt werden, um ein automatisches histologisches Grading durchzuführen. Ein solches, vom Computer durchgeführtes Grading ist objektiv und relativiert die subjektive Merkmalsselektion eines Pathologen. Erfahrene Pathologen können etwa 20–40% der untersuchten Fälle eindeutig klassifizieren, jedoch liegen 60–80% der Fälle zwischen zwei Malignitätsgraden und lassen sich nicht genau zuordnen. Durch eine computergestützte Analyse läßt sich die Aussage eines Pathologen präzisieren.

### **Ausblick**

Die Informatik ist in den letzten 30 Jahren den Weg vom hostbasierten System zum Client/Server-System gegangen. Mit der Entwicklung des World Wide Webs findet der Übergang zu webbasierten Client/Server-Systemen, dem sog. *Netcentric Computing*, statt.

Ein Beispiel für ein Netcentric Computing System ist das im Rahmen dieser Arbeit entstandene WebTed-System. Das WebTed-System erlaubt es als erstes, alle drei Aspekte der Telepathologie über das Web anzubieten. Aufgrund des webbasierten Ansatzes bietet sich eine neue Art der Zusammenarbeit zwischen Wissenschaftlern: Neben der Telediagnostik bietet die quantitative Telepathologie die Chance eine Community aufzubauen, in

der Forschungseinrichtungen ihr Bildverarbeitungs-Know-How in Form von Bildverarbeitungsmodulen auf einem Bildverarbeitungsserver zur Verfügung zu stellen, der in das WebTed-System integriert ist. Somit sind Bildverarbeitungsmodule innerhalb der Community zugänglich und können von den Mitgliedern in ihre visuellen Programme integriert werden.

Damit läßt sich folgendes Szenario realisieren: Auf dem WebTed-Server sind für die Bildverarbeitung Module von verschiedenen Forschungseinrichtungen verfügbar: Forschungseinrichtung X stellt Module für Bildfilterung, Forschungseinrichtung Y für Segmentierung und Forschungseinrichtung Z für Visualisierung bereit.

Für eine Studie möchte ein Wissenschaftler quantitative Parameter aus Bilddaten extrahieren. Der Wissenschaftler nimmt zu verschiedenen Fällen Bilddaten auf und legt diese auf dem WebTed-Server ab. Der Wissenschaftler sichtet die Daten und erstellt aus verfügbaren Modulen ein visuelles Programm. Dazu verwendet er von Forschungseinrichtung X ein spezielles Verfahren zur Bildfilterung, von Y ein Segmentierungsverfahren und kombiniert es mit einem eigenen Verfahren zur Parameterextraktion. Mit dem Verfahren werden alle verfügbaren Fälle analysiert. Die Ergebnisse sehen vielversprechend aus, so daß beschlossen wird, die Resultate in einer größeren klinischen Studie zu verifizieren. Um eine möglichst große Datenbasis für die Verifizierung zu erhalten, werden Kollegen des Wissenschaftlers gebeten, vergleichbare Fälle auf dem WebTed-Server abzulegen.

Der Wissenschaftler arbeitet inzwischen an einer anderen Forschungseinrichtung. Sein Nachfolger ist deshalb mit der Durchführung einer größeren Studie beauftragt. Aufgrund des visuellen Programmieransatzes versteht dieser das Konzept der Bildverarbeitung sehr schnell, optimiert die Bildverarbeitung, so daß die einzelnen Schritte zuverlässig mit allen verfügbaren Daten arbeiten und konzentriert sich auf die Entwicklung statistischer Klassifikationsverfahren, um aus den quantitativen Ergebnissen der Bildverarbeitung einen Malignitätsgrad zu ermitteln. Nachdem ein geeigneter Klassifikator gefunden wurde, wird eine Dokumentation erstellt, in der spezifiziert wird für welche Art von Fällen das Verfahren geeignet ist, und das Verfahren wird auf dem WebTed-Server publiziert. Dort kann es dann von allen Mitgliedern

der Community verwendet werden.

Das oben skizzierte Szenario zeigt, welches Potential das Web und Java für die für medizinische Anwendungen bietet. Es ist daher zu hoffen, daß die Potentiale erkannt und genutzt werden, so daß die Ergebnisse öffentlich geförderter Forschung künftig Anwendung im medizinischen Alltag findet.



# Kapitel 8

## Zusammenfassung

In der vorliegenden Arbeit wurde das WebTed-System für webbasierte Tele-diagnostik vorgestellt. Es unterstützt drei Aspekte der Telepathologie: statische, dynamische und quantitative Telepathologie.

Statische Telepathologie wird für Expertenkonsultationen eingesetzt. Mit einem Webbrowser werden klinische Daten und Bilder zu einem Fall durch einen Referenten auf einem Server abgelegt. Ein auf dem Server abgelegter Fall wird entweder durch einen einzelnen Konsultanten befundet oder durch mehrere Experten gemeinsam in einer Konferenzsitzung.

Nachteil der statischen Telepathologie ist die feste Vorgabe der Ausschnitte eines Präparats. Diesen Nachteil behebt die dynamische Telepathologie. Der Konsultant bedient ein ferngesteuertes Mikroskop und hat selber die Kontrolle über die Bildselektion. Dabei lassen sich alle wesentlichen Funktionen des Mikroskops wie z.B. Fokus, Objektiv und Licht ferngesteuert bedienen.

Zweck der quantitativen Telediagnostik ist die Bestimmung von Parametern zur Beschreibung Gewebestrukturen wie z.B. der Zellmorphologie (Form, Struktur, Verteilung, Häufigkeit) oder molekularbiologischen Parametern (Aktivität des H19 Gens) die zu einer Objektivierung der Diagnose beitragen: Die Bestimmung dieser Parameter wird vollautomatisch vom Computer durchgeführt und ist deshalb nicht durch die Subjektivität eines Pathologen beeinflussbar.

Als Client wird ein vom Webserver heruntergeladenes Java Applet verwendet, mit dem der Benutzer Zugriff auf die statische, dynamische und quantitative Telepathologie hat. Auf dem Server wird eine SQL-Datenbank zur Speicherung von Daten zu Bildern und Fällen eingesetzt, ein Konferenzserver koordiniert den Datenaustausch zwischen den Clients während der Onlinekonsultationen, ein CORBA ORB stellt ein Modul zur Fernsteuerungen eines Mikroskops für die dynamische Telepathologie bereit und die Bildverarbeitungs-komponente wird für die quantitative Analyse der Bilddaten im Rahmen der quantitativen Telepathologie eingesetzt. Den Anforderungen des Bundesdatenschutzgesetzes wird durch Verschlüsselung der Daten mit SSL genüge getan.

# Anhang A

## Übersicht über den Server

Das Client Applet verwendet für die statische, dynamische und quantitative Telepathologie Dienste auf dem Server. Dies sind für die statische Telepathologie HTTP-, Datenbank- und Konferenzdienste, für die dynamische Telepathologie HTTP-Dienste sowie Dienste zur Mikroskopsteuerung und für die quantitative Telepathologie Datenbank- und Bildverarbeitungsdienste. Datenbank, Bildverarbeitung und Mikroskopsteuerung wurden mit CORBA oder Java RMI realisiert. Über den CORBA bzw. Java ORB erhält der Client eine Referenz auf die Serverobjekte. Datenbank- und Bildverarbeitungsdienste wurden mit einer 3-Tier Client/Server-Architektur implementiert, Konferenzdienste und Ansteuerung des Mikroskops wurden mit einer 2-Tier Client/Server-Architektur realisiert.

Abbildung A.1 gibt eine Übersicht über den Server. Dieser gliedert sich in Basisdienste (HTTP, SSH), Remote Objects für Datenbank, Bildverarbeitung und Mikroskopsteuerung und Backennddienste. Der Client hat nur Zugriff auf die Basisdienste und Remote Objects, der Zugriff auf die Backennddienste wird durch die Remote Objects zur Verfügung gestellt. Es stehen folgende Dienste zur Verfügung:

**HTTP:** Der Webserver stellt Webseiten, Bilder und insbesondere die Java Applets für den Web-Client bereit. Als Server wird ein Apache 1.3.4 unter Linux verwendet. Für cgi-bin Erweiterungen wird Python verwendet.

**RMI-, CORBA-, JSDT-ORB:** Die Requestbroker stellen dem Java Applet Referenzen auf unterschiedliche Serverobjekte zur Verfügung. Dies sind das Datenbankobjekt, das Mikroskopobjekt, das Serverobjekt zum Bildverarbeitungsserver und ein Objekt zum Konferenz-Server.

**Bildarchiv:** Zur Speicherung von klinischen Daten, Bildern und den Ergebnissen der quantitativen Telepathologie wird eine Datenbank verwendet. Der Client erhält über den Java RMI-ORB eine Referenz auf das Datenbankobjekt. Das Datenbankobjekt stellt Methoden zum Erzeugen, Löschen etc. von Datensätzen bereit.

Für das Bildarchiv wird eine 3-Tier Client/Server-Architektur verwendet. Der Client kommuniziert über RMI mit dem Serverobjekt. Das Serverobjekt führt die eigentlichen Operationen auf der SQL-Datenbank (Adabas) aus.

**Konferenz:** Das Konferenzmodul stellt für Konferenzsitzungen Mechanismen zum Datenaustausch zwischen mehreren Clients bereit. Das Whiteboard verwendet diesen Dienst, um Einladungen für eine Konferenz an andere Mitglieder zu verschicken oder die Position von Pfeilen zu erhalten bzw. zu verschicken.

**Mikroskop:** Der Client der dynamischen Telepathologie verwendet zur Fernsteuerung der einzelnen Funktionen (Fokus, x-y Tisch, Objektiv, Licht) des Mikroskops ein CORBA-Objekt. Die Bilddaten der am Mikroskop angeschlossenen Videokamera werden per HTTP übertragen.

**Bildverarbeitung:** Der Bildverarbeitungsserver stellt für die quantitative Telepathologie Dienste zur Filterung, Segmentierung, Parameterextraktion etc. von Bilddaten zur Verfügung. Dazu wird eine 3-Tier Client/Server-Architektur verwendet. Anfragen von Clients an den Server werden vom Server (Tier 2) zur Ausführung an einen Bildverarbeitungsserver weitergeleitet.

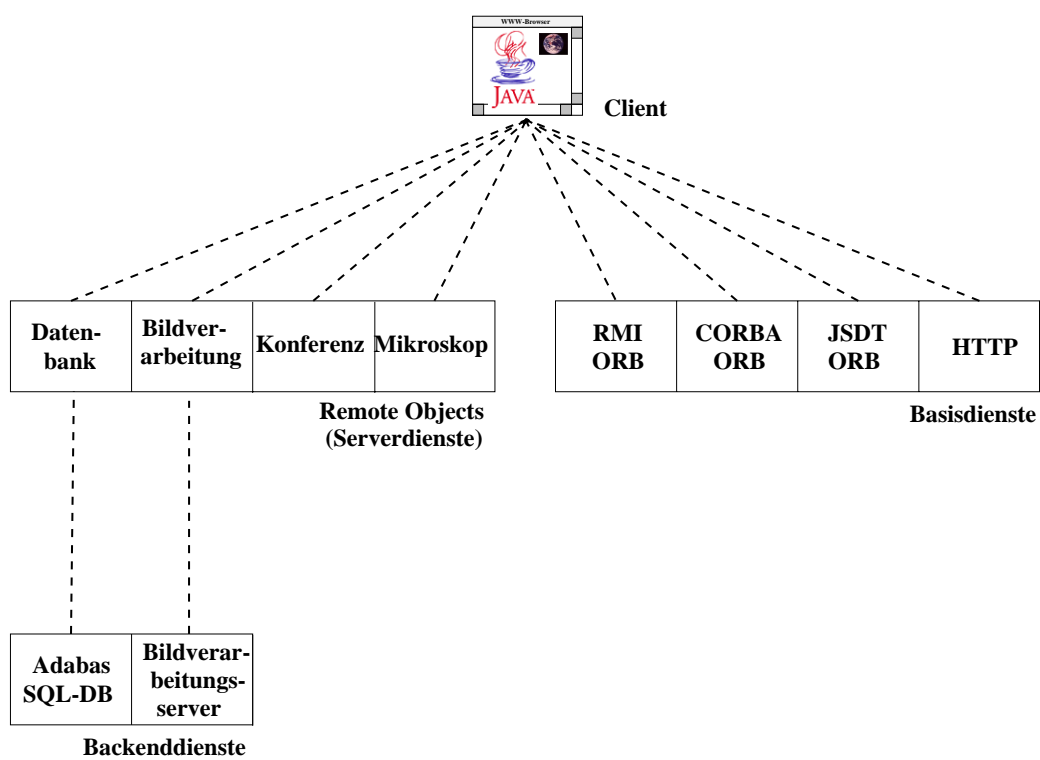


Abbildung A.1: *Serverkomponenten des Systems.*



## Anhang B

# Implementierte Glyphen

In verschiedenen am DKFZ durchgeführten Diplom- und Doktorarbeiten wurden in der Vergangenheit Werkzeuge und Verfahren zur die Verarbeitung histologischer Bilddaten entwickelt. Als Bilddaten wurden mit einem Lichtmikroskop aufgenommene 2D-Bilder von histologischen Schnitten oder mit einem konfokalen Laser Scanning Mikroskop aufgenommene 3D-Bilder verwendet [Geiger 92] [Zhao 94] [Bracht 95] [Janson 96] [Fischer 96] [Bischoff 98] [Kattit 99].

In diesen Arbeiten ist immer wieder das gleiche Muster für die Bestimmung von Parametern zu erkennen: In einem Vorverarbeitungsschritt werden Bilddaten gefiltert, anschließend segmentiert, relevante Objekte selektiert und Objektparameter gemessen. Für diese Operationen wurden z.T. bekannte Bildverarbeitungsoperationen verwendet; z.T. wurden Verfahren für eine spezielle Problemstellung neu entwickelt. Anstelle einer Einführung in die digitale Bildverarbeitung, wie sie z.B. in [Lehmann 97] zu finden ist, soll das Schwergewicht auf den für die Histologie relevanten Verfahren liegen. In den nächsten Abschnitten werden deshalb Glyphen zur Filterung, Segmentierung und Berechnung von Parametern vorgestellt. Neben diesen Glyphen gibt es noch eine große Anzahl weiterer Glyphen, die nur bekannte Standardverfahren implementieren und deshalb nicht vorgestellt werden. Insgesamt sind z.Zt. etwa 50 Glyphen für unterschiedliche Aufgaben implementiert.

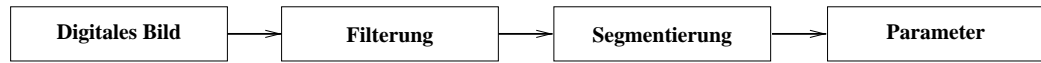


Abbildung B.1: *Notwendige Verarbeitungsschritte zur Berechnung quantitativer Parameter: Filterung, Segmentierung, Parameterberechnung.*

## B.1 Datenarchiv

Am Anfang und am Ende der Verarbeitungspipeline steht der Abruf der Bilddaten aus der Datenbank bzw. die Speicherung der Ergebnisse (Bilder und numerische Daten) in der Datenbank. Dazu gibt es zwei Glyphen: einen *GetImage*- und einen *PutImage*-Glyphen. Mit dem *GetImage*-Glyphen werden Bilder abgerufen und als TIFF-Bild zur Verarbeitung bereitgestellt. Über das Parameterfenster läßt sich die Bildentität auswählen; zu 2D-Zellkern-Bildern gibt es z.B. noch eine Maske für die Segmentierung. Die Archivierung der Ergebnisse erfolgt mit dem *PutImage*-Glyphen. Bis zu zwei Bilder mit zugehörigen quantitativen Daten werden mit diesem Glyphen in der Datenbank abgelegt.

## B.2 Rauschfilterung

Ziel der Bildfilterung ist die Kompensation von systematischen Fehlern der Bildaufnahme. Bei den histologischen Bilddaten geht es dabei in erster Linie um die Rauschunterdrückung, um eine anschließende Segmentierung zu erleichtern. Für die Filterung stehen Glyphen zur Gauß-, Median- und Waveletfilterung zur Verfügung. Gaußfilter sind lineare Filter, die hohe Frequenzanteile wegfiltern und so das Bild glätten. Medianfilter sind nichtlineare Rangordnungsfiler. Aus Pixeln in der Umgebung des zu bearbeitenden Pixels werden Grauwerte der Größe nach sortiert. Das bearbeitete Pixel wird durch dasjenige Pixel, das in der Anordnung in der Mitte steht, ersetzt. Waveletfilter transformieren das Bild in den Waveletraum, setzen alle Koeffizienten unterhalb eines Thresholds auf 0 und führen eine Rücktransformation durch [Donoho 92].



## B.3 Segmentierung

Einer der wichtigsten Verarbeitungsschritte ist die Segmentierung. Zweck der Segmentierung ist die Aufteilung des Bildes in semantisch sinnvolle Regionen, die den in der Szene vorhandenen Objekten entsprechen. Dazu werden Pixel aufgrund von Merkmalen wie Intensität, Farbe, Umgebung zusammengefaßt und einem Objekt zugeteilt. Für die Segmentierung histologischer Bilder stehen Glyphen für punktorientierte, regionenorientierte und kantenorientierte Segmentierungsverfahren zur Verfügung.

Punktorientierte Verfahren treffen die Entscheidung, ob ein Pixel zu einem Objekt gehört aufgrund des Grauwerts des Pixels und versuchen, einen Schwellwert zu finden, der die Pixel in zwei möglichst unterschiedliche Klassen einteilt. Dieses Verfahren wird global, d.h. über das ganze Bild, durch den Glyph *Threshold2D* und lokal adaptiv für die Segmentierung mehrerer kleiner Objekte durch den Glyphen *i3locexl* implementiert.

Regionenorientierte Verfahren berücksichtigen zusätzlich zum Wert eines Pixels die Nachbarschaftsbeziehungen einer Region. Dem Zusammenhang von benachbarten Pixeln wird dadurch Rechnung getragen. Ein solches Verfahren wird durch den Glyphen *ColorSeg* bereitgestellt, der das von Meer entwickelte Verfahren implementiert [Meer 99]. Es liefert für farbige Histologiebilder ausgezeichnete Ergebnisse und findet in Abschnitt 6.6.3 Anwendung.

Kantenorientierte Verfahren verfolgen Linien entlang starker Änderungen der Intensitätswerte eines Bildes. Zweidimensionale histologische Bilder weisen neben den Zellkernen auch kontrastreiche Gewebestrukturen auf, die zu Fehlsegmentierungen führen. Kantenorientierte Verfahren dienen deshalb nur zur Verfeinerung einer Grobsegmentierung, die z.B. mit einem Threshold-Verfahren gewonnen wurde. Die Feinsegmentierung wird mit aktiven Konturmodellen (Snakes) durchgeführt. Das Segmentierungsproblem wird als pseudo-physikalisches Energieminimierungsproblem formuliert: entlang der Objektkontur, die als Splinefunktion gegeben ist, wird eine Energiegleichung formuliert. In einem Optimierungsschritt werden die Stützstellen so gewählt, daß die Energie entlang der Kontur minimiert wird. Der Glyph *Snake* ver-

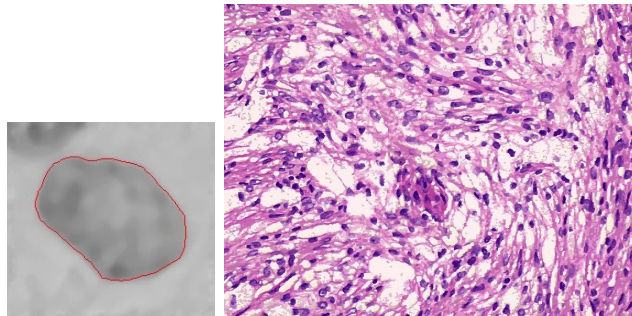


Abbildung B.2: *Beispiele zur Segmentierung. Links: der Glyph Snake minimiert das Integral über eine Energiefunktion entlang der Kontur. Rechts: Farbsegmentierung mit dem Glyphen ColorSeg. Das Ursprungsbild mit 107.000 Farben wurde auf 20 segmentierte Farbenklassen reduziert. Für Zellkerne werden üblicherweise 2-4 Farben benötigt.*

wendet das in [Lai 94] entwickelte Verfahren. Es eignet sich gut für die Optimierung der Grobsegmentierung von Graustufenbildern einzelner Zellkerne.

## B.4 Datenanalyse

Die Segmentierung liefert als Ergebnis Bilder, in denen die einzelnen Objekte markiert sind. Der Analyseschritt berechnet aus den Objektdaten Merkmale, die die Beziehungen der Objekte zueinander quantitativ charakterisieren. Je nach gestellter Aufgabe werden unterschiedliche Merkmale benötigt. Die Berechnung der Merkmale erfolgt wiederum mit Python. Die notwendigen Mathematikmodule werden durch die Erweiterung *Numerical Python* bereitgestellt [Numerical Python]. An dieser Stelle werden einige Merkmale für 2D- und 3D-Bilder vorgestellt, die im Abschnitt Anwendungen verwendet werden.

### B.4.1 2D-Merkmale

Zur Klassifizierung von Zellkernen werden im zweidimensionalen Fall Merkmale wie Fläche, Objektkontur, Orientierung und Objektverbund verwendet.

### Flächenberechnung

Das einfachste Merkmal stellt die Fläche eines Zellkerns dar, es berechnet sich durch die Anzahl der Pixel, multipliziert mit der Größe eines Pixels:

$$KF = \text{Pixelanzahl} * \text{Pixelfläche}$$

### Konturlänge

Die Länge des Kernumfangs ergibt sich aus der Konturkodierung nach Freeman [Lehmann 97]. An der Kontur entlang wird das Objekt umlaufen und bestimmt, in welcher Richtung sich der nächste Konturpunkt befindet. Bei der verwendeten 8er Nachbarschaft kann der nächste Punkt nur in einer von 8 möglichen Richtungen im quadratischen Raster liegen (Abb. B.3). Die Richtungen werden mit den Zahlen 0 bis 7 kodiert. Eine Kontur läßt sich bei gegebenem Startpunkt und Startrichtung komplett durch eine Sequenz der Richtungen, den Konturcode, beschreiben. Die Länge einer Kontur ergibt sich durch Summation über den Konturcode, wobei Richtungen mit geradem Code die Länge 1, und Richtungen mit ungeradem Code die Länge  $\sqrt{2}$  besitzt.

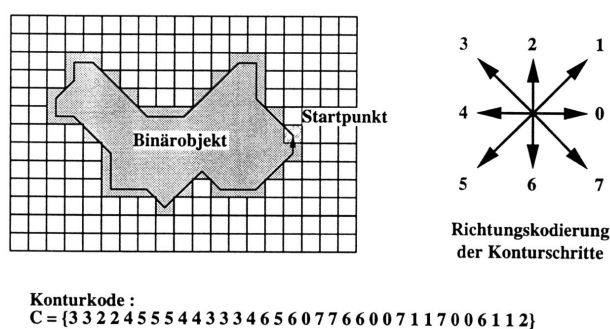


Abbildung B.3: *Prinzip des Kettencodes nach Freeman (aus [Fischer 96]).*

### Formfaktor

Der Formfaktor beschreibt das Verhältnis von Umfang und Fläche eines Zellkerns. Für runde Zellkerne beträgt der Formfaktor 1, je unregelmäßiger die

Kontur, d.h. je länger die Kontur wird, desto kleiner wird der Formfaktor. Der Formfaktor ist ein Maß für die Rundheit eines Zellkerns. Er berechnet sich durch

$$RND = \frac{4\pi\text{Fläche}}{\text{Konturlänge}^2}$$

### Ellipsenanpassung

Für die Berechnung weiterer Merkmale des Zellkerns wird eine Ellipsenanpassung an den Zellkern vorgenommen. Die Ellipse wird dabei als Idealkontur für einen Zellkern angenommen, aus dem die Verhältnisse der Form und die Orientierung des Zellkerns berechnet werden. Dazu wird das aus der linearen Algebra bekannte Verfahren der Hauptachsentransformation verwendet. Es ist in [Kett 90] beschrieben. Aus der Ellipsenanpassung gehen die Güte der Ellipsenanpassung

$$GE = \frac{1}{N} \sum_{n=1}^N \sqrt{(x_{\text{Ellipse}} - x_{\text{Zellkern}})^2 + (y_{\text{Ellipse}} - y_{\text{Zellkern}})^2}$$

sowie mit der großen Halbachse  $a$  und der kleinen Halbachse  $b$  die Exzentrizität der Ellipse

$$EXZ = \sqrt{1 - \frac{b^2}{a^2}}$$

hervor.

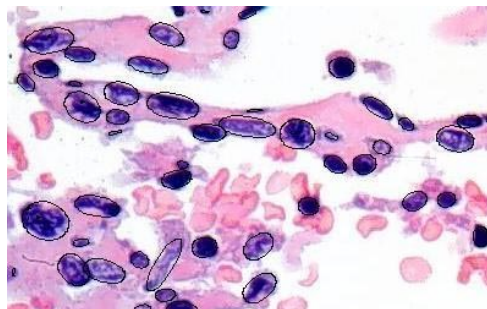


Abbildung B.4: *Ellipsenanpassung. An jeden segmentierten Zellkern wird eine Ellipse angepaßt, um die Orientierung zu bestimmen.*

### Polarität

In gesundem Gewebe sind Zellkerne regelmäßig ausgerichtet, im Falle einer Neoplasie ist diese Ordnung gestört. Diese läßt sich durch die relativen Ausrichtungen der Ellipsen untereinander parametrisieren. Als Maß wird die mittlere Differenz zu den Nachbarn verwendet:

$$POL = \frac{1}{N} \left( \sum_{n=1}^N |\phi - \phi_n| \right) \bmod \frac{\pi}{2}$$

### Voronoi Partitionen

Für die Charakterisierung der räumlichen Organisation der Zellen werden Voronoi Diagramme verwendet. Diese berechnen zu jedem Punkt einer Punktmenge  $S = \{p_1, p_2, p_3, \dots\}$ , Voronoi Regionen. Die Voronoi Regionen haben die Eigenschaft, daß alle Punkte innerhalb der Voronoi Region  $V_j$  einen kleineren Abstand zu  $p_j$  haben als zu irgendeinem anderen Punkt  $p_i$ .

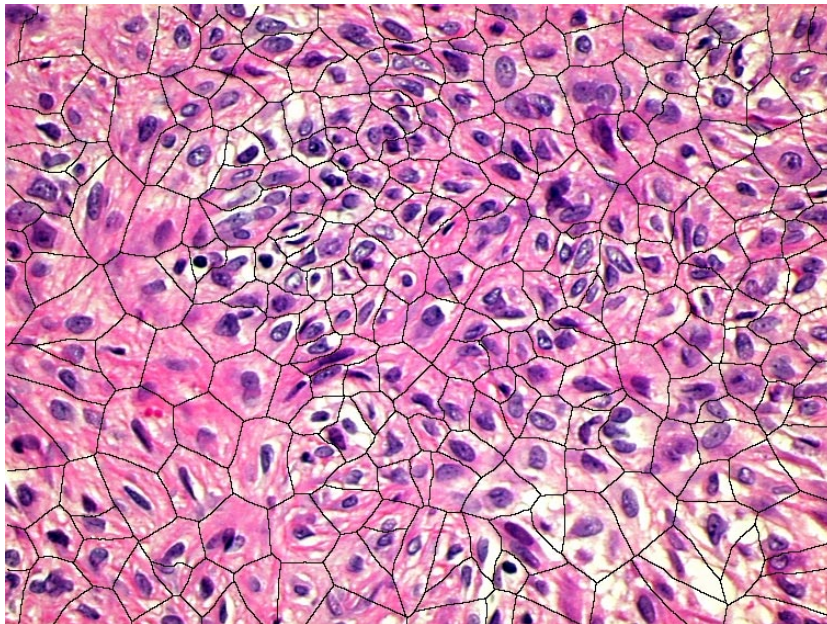


Abbildung B.5: *Aufteilung in Voronoi-Regionen.*

Mit der euklidischen Distanzfunktion  $d((x, y), (x_i, y_i))$  ergibt sich für die

Voronoipartition  $V_i$  zu einem Punkt  $p_i$

$$V_i = \{(x, y) \in \text{Bild} \mid d((x, y), (x_{p_i}, y_{p_i})) < d((x, y), (x_{p_j}, y_{p_j}))\}, \forall i \neq j$$

Für den Einsatz in der Histologie werden anstelle der Schwerpunkte der Zellkerne (Abb. B.5) die Konturen der Zellkerne verwendet, um eine bessere Anpassung der Partitionen an die segmentierten Zellkerne zu erreichen. Mit den Zellkernregionen  $K_1, K_2, \dots$  bedeutet dies, daß ein Pixel derjenigen Region zugeordnet wird, die einen kleineren Abstand zu einem Pixel der Kontur  $K_i$  als zu den Pixeln der anderen Konturen hat:

$$V_i = \{(x, y) \in \text{Bild} \mid d((x, y), K_i) < d((x, y), K_j)\}, \forall i \neq j$$

mit den Punkten  $(x_k, y_k)$  der Kontur  $K_i$  und der Distanzfunktion

$$d((x, y), K_i) = \min\{\sqrt{(x - x_k)^2 + (y - y_k)^2}\}, k = 1, 2, 3, \dots$$

### B.4.2 3D-Merkmale

Mit einem konfokalen Laser Scanning Mikroskop werden dreidimensionale Bilder von Zellkernen aufgenommen. Es handelt sich dabei um Zweikanalaufnahmen, bei denen mit dem einen Kanal der Zellkern, und mit dem zweiten Kanal z.B. Kinotechoren aufgenommen werden.

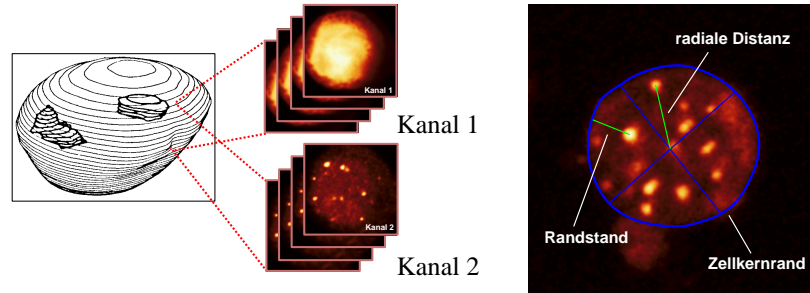


Abbildung B.6: 3D Datensatz eines Zellkerns mit 2 Kanälen (aus [Bischoff 98]).

Bei dieser Art von Datensätzen sind charakteristische Merkmale die Anzahl, Größe und Position der Kinotechoren und die Größe des Zellkerns. Es wurde ein Glyph entwickelt, um die in [Bischoff 98] beschriebenen Parameter zu berechnen. Dieser Glyph ermittelt den Abstand der Kinotechoren zum Zentrum des Zellkerns und den Abstand zum Rand des Zellkerns (siehe Abschnitt 6.6.1).





# Anhang C

## Sicherheitskonzept

Die Verarbeitung personenbezogener Daten unterliegt in Deutschland besonderen gesetzlichen Bestimmungen, die im Bundesdatenschutzgesetz (BDSG) geregelt sind. Ein funktionierendes Sicherheitskonzept ist deshalb von zentraler Bedeutung für den praktischen Einsatz der Telepathologie. Dies gilt insbesondere, wenn für die Datenübermittlung unsichere Kanäle wie das Internet verwendet werden, so daß Daten bei der Übertragung vom Sender zum Empfänger mit relativ einfachen Mitteln abgehört und manipuliert werden können.

Eine absolut sichere Methode, Abhören und Manipulation zu verhindern gibt es nicht, allerdings bietet die Kryptographie Möglichkeiten, diesen Vorgang für einen Angreifer erheblich zu erschweren.

## Anforderungen nach BDSG

Zwar werden von der AG-KT prinzipiell nur anonymisierte Daten verwendet, aber es gibt dennoch Gründe, die eine Verschlüsselung der klinischen Daten erforderlich machen: Aus Kombinationen des Einsenders eines Falles, des Geschlechts und des Alters der Patienten können indirekte Schlüsse gezogen werden. Der Datenschutz richtet sich nach dem Bundesdatenschutzgesetz (BDSG), dabei ist insbesondere die Umsetzung von §9 relevant:

## § 9 Technische und organisatorische Maßnahmen

*Öffentliche und nicht-öffentliche Stellen, die selbst oder im Auftrag personenbezogene Daten verarbeiten, haben die technischen und organisatorischen Maßnahmen zu treffen, die erforderlich sind, um die Ausführung der Vorschriften dieses Gesetzes, insbesondere die in der Anlage zu diesem Gesetz genannten Anforderungen, zu gewährleisten. Erforderlich sind Maßnahmen nur, wenn ihr Aufwand in einem angemessenen Verhältnis zu dem angestrebten Schutzzweck steht.*

In der Anlage werden die einzelnen zu beachtenden Punkte geregelt, die Anforderungen und die Implementierung ihrer Lösung sind in der Tabelle beschrieben.

	Anforderung nach §9	Lösung
1	Unbefugten den Zugang zu Datenverarbeitungsanlagen, mit denen personenbezogene Daten verarbeitet werden, zu verwehren (Zugangskontrolle).	Der Server mit den Daten steht in einem Bereich, der nur Mitarbeitern der Zentralen Datenverarbeitung des DKFZ zugänglich ist.
2	Zu verhindern, daß Datenträger unbefugt gelesen, kopiert, verändert oder entfernt werden können (Datenträgerkontrolle).	Der Zugang zu den Datenträgern ist durch die obige Zugangskontrolle und durch das Betriebssystem des Servers (Linux) gesichert. Das Backup für den Rechner wird zentral vom DKFZ durchgeführt, zum Backup haben nur der Administrator des Rechners und die Administratoren des Backups Zugang.

	Anforderung nach §9	Lösung
3	Die unbefugte Eingabe in den Speicher sowie die unbefugte Kenntnisaufnahme, Veränderung oder Löschung gespeicherter personenbezogener Daten zu verhindern (Speicherkontrolle).	Es wird das Sicherheitssystem der Datenbank verwendet (erfüllt C2-Standard). Dieses erlaubt es nur angemeldeten Benutzern, denen entsprechende Privilegien in der Datenbank verliehen sind, Veränderungen durchzuführen. Der Datenbankadministrator (DBA) darf sich nicht über das Java Applet anmelden.
4	Zu verhindern, daß Datenverarbeitungssysteme mit Hilfe von Einrichtungen zur Datenübertragung von Unbefugten genutzt werden können (Benutzerkontrolle).	Der Server ist durch eine Firewall so konfiguriert, daß nur auf die für die Telepathologie benötigten Dienste zugegriffen werden kann, alle anderen Dienste (NFS, Telnet,...) stehen nicht zur Verfügung. Für die Wartung wurde ein Secure Shell (ssh) Zugang offengehalten, der von nur einem Rechner zugänglich ist.
5	Zu gewährleisten, daß die zur Benutzung eines Datenverarbeitungssystems Berechtigten ausschließlich auf die ihrer Zugriffsberechtigung unterliegenden Daten zugreifen können (Zugriffskontrolle).	Es werden die Sicherheitsfeatures der Datenbank verwendet. Benutzer bekommen innerhalb der DB nur eine Sicht (View) auf die Daten, zu denen sie Berechtigungen haben.
6	Zu gewährleisten, daß überprüft und festgestellt werden kann, an welche Stellen personenbezogene Daten durch Einrichtungen zur Datenübertragung übermittelt werden können (Übermittlungskontrolle).	Transfers werden in einer Logdatei mitgeschrieben.

	Anforderung nach §9	Lösung
7	Zu gewährleisten, daß nachträglich überprüft und festgestellt werden kann, welche personenbezogenen Daten zu welcher Zeit von wem in Datenverarbeitungssysteme eingegeben worden sind (Eingabekontrolle).	Die hierzu benötigten Daten werden mitprotokolliert.
8	Zu gewährleisten, daß personenbezogene Daten, die im Auftrag verarbeitet werden, nur entsprechend den Weisungen des Auftraggebers verarbeitet werden können (Auftragskontrolle).	Ausgeschlossen, da nur der Eigentümer der Daten Änderungsrechte hat. Andere Benutzer können nur Datensätze anlegen, die sich darauf beziehen.
9	Zu verhindern, daß bei der Übertragung personenbezogener Daten sowie beim Transport von Datenträgern die Daten unbefugt gelesen, kopiert, verändert oder gelöscht werden können (Transportkontrolle).	Es wird eine Verschlüsselung gemäß den Empfehlungen des BSI zum Signaturgesetz eingesetzt. Bis zum Jahr 2004 hält die BSI eine Schlüssellänge von mindestens 128 Bit für sicher.
10	Die innerbehördliche oder innerbetriebliche Organisation so zu gestalten, daß sie den besonderen Anforderungen des Datenschutzes gerecht wird (Organisationskontrolle).	Diese Anforderung gilt auch für die gewöhnliche Konsultation, wird also bereits erfüllt.

# Anhang D

## IDL-Mikroskop Interface

**Aktuelle x,y,z Position abfragen:**

`long getXPos(in string id)`

`long getYPos(in string id)`

`long getZPos(in string id)`

**Bestimmte (x,y) Position anfahren:**

`void setXYPos(in string id,`

`in long x, in long y)`

**Intensität der Lampe abfragen:**

`long getLight(in string id)`

**Intensität der Lampe einstellen:**

`void setLight(in string id,`

`in long val)`

**Maximalen Pixelwert im letzten Bild abfragen:**

`long getIntensity(in string id)`

**Joystick für manuelle Steuerung ein- oder ausschalten:**

`void setJoystick(in string id,`

`in long state)`

**Joystick Status abfragen:**

`long getJoystick(in string id)`

**Auf eine Position fokussieren:**

`long focusAt(in string id,`

`in long x, in long y, in long z,`

`in long stop, in long speed)`

**Maß für den Autofokus abfragen:**

`double getFocus(in string id)`

**Aktuelles Bild als JPEG speichern:**

`string grab(in string id,`

`in string name,in long x,`

`in long y, in long z,in long obj,`

`in long light)`

**Verwendetes Objektiv abfragen:**

`long getObj(in string id)`

**Verwendetes Objektiv ändern:**

`void setObj(in string id,`

`in long number)`



# Anhang E

## Glossar

**AG-KT:** Arbeitsgemeinschaft Knochentumoren am Deutschen Krebsforschungszentrum.

**API:** Application Programming Interface.

**CORBA:** Common Object Request Broker.

**CT:** Computertomographie.

**DAL:** Database Abstraction Layer.

**DBMS:** Database Management System.

**DI:** Dataitem, entspricht einem Bild, das zu einem Fall gehört.

**DICOM:** Digital Imaging and Communication in Medicine.

**DS:** Dataset, entspricht einem Fall in der Pathologie. Ein Fall der AG Knochentumoren enthält z.B. klinische Informationen, Einsender und gemeinsame Diagnose der AG.

**EMACS:** Eight Megabytes Continuously Swapping.

**E-Services:** Unter E-Services werden elektronische Dienste verstanden, die ein Benutzer über das Web nutzen kann, ohne daß eine Person direkt in der Erbringung dieser Dienste involviert ist.

**HTML:** Hypertext Markup Language.

**HTTP:** Hypertext Transfer Protocol.

**JFC:** Java Foundation Classes.

**JMF:** Java Media Framework.

**JNI:** Java Native Interface. Schnittstelle, um von Java aus C/C++ Bibliotheken aufzurufen.

**JSDT:** Java Shared Data Toolkit.

**JVM:** Java Virtual Machine.

**Linux:** Das Penguin Betriebssystem von Linus Torvaldis. Pinguine fliegen nicht und können deshalb nicht abstürzen (→Windows).

**ORB:** Object Request Broker. Der Object Request Broker ermöglicht die Kommunikation der Objekte des Anwendungssystems. Ein Client wendet sich zunächst an den Broker, der einen geeigneten Server für die gewünschte Dienstleistung ermittelt und den Aufruf transparent für den Client ausführt.

**RFC:** Request for Comments.

**RMI:** Remote Method Invocation.

**Shared Library:** Programmbibliothek, die dynamisch zu einem Programm gelinkt wird und von mehreren Programmen verwendet wird.

**SSI:** Server Side Includes.

**SSL:** Secure Socket Layer. Transportschicht zur Datenübertragung, welche die Verschlüsselung übernimmt, um sensible Informationen sicher verbreiten zu können.

**SWING:** Javabasiertes Fenster Toolkit, in den JFC enthalten.

**WFC:** Windows Foundation Classes.



**Windows:** Betriebssystem aus Redmond mit fliegenden Fenstern. Fliegt und ist deshalb akut absturzgefährdet ( $\rightarrow$ Linux).



# Literaturverzeichnis

[AVS] *Advanced Visual Systems.*

<http://www.avs.com>

[Adabas] *Adabas the ultimate fast data management server, - relational and beyond.*

<http://www.adabas.com/>

[Allen 93] A. Cypher (Hrsg.): *Watch What I Do. Programming by Demonstration.* MIT Press Cambridge, Massachusetts London, England, 1993.

[Ariel 95] I. Ariel, O. Lustig, T. Schneider, G. Pizkov, M. Sappir, N. de Groot, A. Hochberg: *The Imprinted H19 Gene as Tumor Marker in Bladder Carcinoma.* In: *Urology*, Vol. 45, 1995, S. 335–338.

[BSI] *Bundesamt für Sicherheit in der Informationstechnik.*

<http://www.bsi.de/>

[Bashshur 75] R. L. Bashshur, P. A. Armstrong, Z. I. Youssef: *Telemedicine; Explorations in the use of telecommunications in health care.* Charles C. Thomas, Springfield, IL, USA, 1975.

[Bashshur 80] R. Bashshur: *Technology serves the people; the story of a cooperative telemedicine project by NASA, the Indian Health Service and the Papago people.* Superintendent of Documents, US Government Printing Office, Washington D.C., USA, 1980.

- [Bischoff 98] A. Bischoff: *Rekonstruktion und 3D-Darstellung der inneren Struktur des Zellkerns mit Hilfe der konfokalen Laser-Scanning-Mikroskopie*. Dissertation, Universität Heidelberg, 1998.
- [Boland 96] G. W. Boland, P. R. Mueller, J. H. Thrall: *Subspecialty Telerradiology: Cost Analysis*. In: Radiology Scientific Program, Radiological Society of North America, 1996, S. 143.
- [Bracht 95] R. Bracht: *Dreidimensionale Analyse der Zellkernstruktur, dargestellt mit dem Konfokalen Fluoreszenzmikroskop*. Dissertation, Universität Heidelberg, 1995.
- [Braine 97] L. Braine, C. Clack: *Object-Flow*. In: IEEE Symposium on Visual Languages, Capri, Italien, September 1997.
- [Buccigrossi 97] R. Buccigrossi, E. Simoncelli. *Progressive wavelet image coding based on a conditional probability model*. In: Proceedings ICASSP 97, April 21-24, München, 1997.
- [Burchert 99] H. Burchert: *Der Einsatz der medizinischen Telekommunikation in der Neurochirurgie aus Sicht der Ökonomie*. In: J. H. Müller (Hrsg.): Medizinische Telekommunikation. Beispiel Neurochirurgie. Berlin, Heidelberg, u.a., 1999, S. 53–85.
- [Byte 6/98] T. Halfhill: *Java — neun Wege aus der Performance Falle*. In: Byte 6/98, Juni 1998, S. 32–43.
- [Callas 97] P. W. Callas, K. O. Leslie, A. R. Mattia, D. L. Weaver, D. Cook, B. Travis, D. E. Stanley, L. A. Rogers, S. L. Mount, T. D. Trainer, M. A. Zarka, R. M. Belding: *Diagnostic accuracy of a rural live video telepathology system*. In: American Journal of Surgical Pathology 21(7), 1997, S. 812–819.
- [CancerNet] *CancerNet.com*.  
<http://www.cancernet.com>

- [Citrin 96] A. Citrin, V. Wayne, C. Santiago, B. Zorn: *Scalable Interfaces to Support Program Comprehension*. In: 1996 IEEE Workshop on Program Comprehension, Berlin, 1996, S. 123–132.
- [Crowther 95] J. B. Crowther, R. Poropatich: *Telemedicine in the U.S. Army: Case reports from Somalia and Croatia*. In: Telemedicine Journal 1(1), 1995, S. 73–80.
- [DICOM 3.0] American College of Radiology, National Electrical Manufacturers Association: *Digital Imaging and Communications in Medicine (DICOM)*. Rosslyn VA: NEMA Publication Sales, 1993.
- [DICOM VL] *Supplement 15 for the DICOM Standard. ACR/NEMA*.
- [Della 98] V. Mea Della, P. Cataldi, S. Boi, N. Finato, P. Dalla Palma, C. A. Beltrami: *Image selection in static telepathology through the Internet*. In: Journal of Telemedicine and Telecare 4 (Suppl 1), 1998, S. 20–22.
- [DevEdge] *DevEdge Online - DevEdge Home Page*.  
<http://developer.netscape.com>
- [Donoho 92] D. L. Donoho, I. M. Johnstone: *Ideal Spatial Adaption by Wavelet Shrinkage*. Technical Report 400, Stanford University, 1992.
- [Dudek 95] *Communication Standards in Healthcare*. In: Prokosch H.U., Dudek J. (Hrsg.): Hospital Information Systems: Design and Development Characteristics, Impact and Future Architecture. Amsterdam: North-Holland, 1995.
- [E J Pathol] *Electronic Journal of Pathology and Histology*. Verlag für angewandte Wissenschaften, München.
- [EUROPATH] *Welcome to the EUROPATH Project*.  
<http://europath.imag.fr/>
- [Eindhoven 1906] W. Eindhoven: *Le telecardiogramme*. In: Arch. Int. Physiol. IV, 1906, S. 132–165.

- [Ernestus 00a] W. Ernestus: *Biometrische Merkmale und Datenschutz*. In: Jäckel (Hrsg.): *Telemedizinführer Deutschland*, Ausgabe 2000, Bad Nauheim, 1999, S. 70–71.
- [Ernestus 00b] W. Ernestus: *Datenschutzrechtliche Aspekte bei der Archivierung auf optischen Datenträgern*. In: Jäckel (Hrsg.): *Telemedizinführer Deutschland*, Ausgabe 2000, Bad Nauheim, 1999, S. 76–80.
- [FHG] *Schlaganfall-Teleservice Saar*.  
<http://www.ibmt.fhg.de/public/Produktblaetter/pages/sts.htm>
- [Falk 99] V. Falk, T. Walter, A. Diegeler, J. Brucerius, F. W. Mohr: *Telemanipulator-Assistierte endoskopische Herzchirurgie*. In: Jäckel (Hrsg.): *Telemedizinführer Deutschland*, Ausgabe 2000, Bad Nauheim, 1999, S. 297–299.
- [Ferrer-Roca 95] O. Ferrer-Roca, A. Ramos, A. Diaz-Cardama: *Inmunohistochemical correlation of steroid receptors and disease-free interval in 206 consecutive cases of breast cancer: validation of telequantification based on global scene segmentation*. In: *Anal. Cell Pathol* 9, 1995, S. 151–163.
- [Fischer 96] M. Fischer: *Entwicklung eines Bildverarbeitungssystems zur quantitativen Analyse der Expression des Gens H19 und des histologischen Malignitätsgrades am Harnblasenkarzinom*. Dissertation, Universität Heidelberg, 1996.
- [Foote 76] D. Foote, H. Hudson, E.B. Parker: *Telemedicine in Alaska: the ATS-6 satellite biomedical demonstration*. National Technical Information Service (NTIS). U.S. Department of Commerce. Springfield, VA, USA, 1976.
- [Frei 78] H.P. Frei, D.L. Willer, R. Williams: *A Graphics-Based Programming Support System*. In: *ACM Computer Graphics (Proceedings SIGGRAPH)*, Vol. 12, No. 3, August 1978, S. 43–49.

- [From 93] S. From, L. A. Stenvold, T. Danielsen: *Telemedicine services integrated into a health care network*. Telektronikk, Norwegian Telecom Research, 89(1), 1993, S. 12–22.
- [Gamma 96] E. Gamma, R. Helm, R. E. Johnson, J. M. Vlissides: *Entwurfsmuster. Elemente wiederverwendbarer objektorientierter Software*. Addison-Wesley, München 1996.
- [Geiger 92] B. Geiger: *Dreidimensionale modellgestützte Bildverarbeitung zur Struktur-Analyse von hochaufgelösten Fluoreszenz-Bildern eines konfokalen Laser-Scanning Mikroskops*. Dissertation, Universität Heidelberg, 1995.
- [Glinert 84] E. P. Glinert, S. L. Tanimoto. *PICT: An Interactive, Graphical Programming Environment*. In: IEEE Computer 17(11):7–25, November 1984.
- [Goetz 00a] C. F. J. Goetz: *Entwicklung des ersten deutschen HPC, dem elektronischen Arztausweis*. In: Jäckel (Hrsg.): *Telemedizinführer Deutschland*, Ausgabe 2000, Bad Nauheim, 1999, S. 85–88.
- [Goetz 00b] C. F. J. Goetz: *TeleTrust: Kryptoreport*. In Jäckel (Hrsg.): *Telemedizinführer Deutschland*, Ausgabe 2000, Bad Nauheim, 1999, S. 95–118.
- [Gong 99] L. Gong: *Inside Java 2 Platform Security: Architecture, API Design, and Implementation*. Addison Wesley, 1999.
- [Good 99] J. Good: *Programming Paradigms, Information Types and Graphical Representations: Empirical Investigations of Novice Program Comprehension*. PhD Thesis, University of Edinburgh, 1999.
- [Gosling 98] R. Menge: *Java One, das Stellwerk für die Zukunft von Java, mit einem Interview von J. Gosling*. In: Byte 6/98, Juni 1998, S. 18–21.

- [Grimes 97] G. Grimes, S. McClellan, J. Goldman, D. A. Conner, S. McClellan, G. Vaughn, T. Winokur, W. Fleming, J. McDonald: *Applications of Virtual Reality Technology in Pathology*. In: Proc. Medicine Meets Virtual Reality 5 San Diego, CA, Februar 1997.
- [Grotehen 95] T. Grotehen, K. Dittrich: *CORBA: Persistente Objekte aus der Steckdose*. In: Datenbank Fokus, Februar/März 1995, S. 41–48.
- [Gyssens 94] M. Gyssens, J. Paredaens, J. Van den Bussche, D. Van Gucht: *A Graph-Oriented Object Database Model*. In: IEEE Transactions on Knowledge and Data Engineering, 6, 1994, S. 572–586.
- [HISTKOM] J. Schmid, P. Schwarzmann, B. Binder, J. Burkart, R. Klose: *Field test to evaluate telepathology with remotely driven microscopy—Project HISTKOM*. In: Cell Vision 3(6), 1996, S. 479–481.
- [Hayes 96] T. Hayes, A. Kinsella, N. A. Brown, D. A. Perednia: *The Telemedicine Information Exchange (TIE)*. In: The Journal of Telemedicine and Telehealth 2(1), 1996.
- [Hungunin 97] J. Hugunin: *Python and Java - The Best of Both Worlds*. In: Proceedings of the 6th International Python Conference, San Jose, California, USA, Oktober 14-17, 1997.
- [JFC Tutorial] K. Walrath, M. Campione: *The JFC Swing Tutorial: A Guide to Constructing GUIs*. Addison-Wesley, 1999.
- [JMF] *Java(TM) Media Framework 2.0*.  
<http://java.sun.com/products/java-media/jmf/2.0/>
- [JPython] *JPython Home*.  
<http://www.jpython.org>
- [JSDT] *Java(TM) Shared Data Toolkit Home Page*.  
<http://java.sun.com/products/java-media/jsdt/index.html>



- [Janson 96] C. P. Janson: *Bildanalysemethoden und mathematische Klassifikationsverfahren zum histologischen Grading des Prostatakarzinoms*. Dissertation, Universität Heidelberg, 1996.
- [Java Tutorial] M. Campione, K. Walrath: *The Java Tutorial Second Edition: Object-Oriented Programming for the Internet (Java Series)*. Addison-Wesley, 1998.
- [Kattit 99] F. Kattit: *Bildverarbeitungsmethoden zur quantitativen Strukturuntersuchung des Zytoskeletts*. Dissertation, Universität Heidelberg, 1999.
- [Kayser 99] K. Kayser, J. Symas, R. Weinstein: *Telepathology*. Springer Verlag, 1999.
- [Kett 90] P. Kett: *Methoden der Bildverarbeitung für die automatische Segmentierung und Strukturanalyse von dreidimensionalen Bildern eines konfokalen Laser Scanning Mikroskops*. Dissertation, Universität Heidelberg, 1990.
- [Khoros] *Khoros Research, Inc.*.  
<http://www.khoros.com>
- [Kimura 86] T. D. Kimura, J. W. Choi, J. M. Mack: *A visual language for keyboardless programming*. Technical Report WUCS-86-6, Department of Computer Science, Washington University, St. Louis, MO, USA, März 1986.
- [Klose 99] B. Binder, P. Schwarzmann, R. Klose: *Interoperability within a European telepathology framework*. In: 6th ESACP Congress, Heidelberg, April 7-11, 1999.
- [Knuth 73] D. E. Knuth: *The Art of Computer Programming: Fundamental Algorithms*. Addison-Wesley, 1973.
- [Kunz 92] C. M. Kunz: *Entwicklung von Methoden zur Erfassung und quantitativen Analyse der Bewegung lebender Zellen*. Dissertation, Universität Heidelberg, 1992.

- [Kunze 98] G. Haroske, W. Meyer, F. Theissig, K. Schubert, K. D. Kunze: *Remote quantitation server for quality assurance in DNA ploidy analysis*. In: Anal Quant Cytol Histol 20(4), August 1998, S. 302–312.
- [LabView] *LabVIEW from National Instruments*.  
<http://www.ni.com/labview/>
- [LINK] *Springer LINK*.  
<http://link.springer.de/>
- [Lai 94] K. Fung Lai: *Deformable Contours: Modelling, Extraction, Detection and Classification*. PhD Thesis, University of Wisconsin-Madison, 1994.
- [Laird 98] C. Laird, K. Soraiz: *Scripts im Griff. Script-Sprachen im Vergleich*. In: Byte 7/98, Juli 1998, S. 56–61.
- [Lehmann 97] T. Lehmann, W. Oberschelp, E. Pelikan: *Bildverarbeitung für die Medizin. Grundlagen, Modelle, Methoden, Anwendungen*. Springer Verlag, 1997.
- [Loane 99] M. Loane, R. Wootton: *The UK Multicentre Teledermatology Trial: results from the first 1000 patients*. In: 1st European Symposium on Teledermatology, November, 1999.
- [Luratech] *LuraTech Home*.  
<http://www.luratech.com>
- [Mairinger 96] T. Mairinger, C. Gabl, P. Derwan, G. Mikuz, O. Ferrer-Roca: *What do physicians think of telemedicine? A survey in different European regions*. In: Journal of Telemedicine and Telecare 2(1), 1996. S. 50–56.
- [Mairinger 98] T. Mairinger, W. Schoner, A. Gshwendtner, T. Netzer: *Pathologists' attitudes to implementing telepathology*. In: Journal of Telemedicine and Telecare 4(1), 1998, S. 41–46.

- [Mattes 98] J. Mattes: *Der Structure Designer. Eine graphische Oberfläche zur Erstellung verteilter, komponentenbasierter Anwendungen*. Diplomarbeit, Universität Heidelberg, 1998.
- [McGraw 96] G. McGraw, E. Felten: *Java Security: Hostile Applets, Holes, and Antidotes*. John Wiley & Sons, Inc., 1996.
- [McGraw 99] G. McGraw., E. Felten: *Securing Java: Getting Down to Business with Mobile Code*. 2nd Edition. John Wiley & Sons, Inc., 1999.
- [Medscape] *Today on Medscape*.  
<http://www.medscape.com/>
- [Meer 97] D. Foran, P. Meer, T. Papathomas, I. Marsic: *Compression Guidelines for Diagnostic Telepathology*. In: IEEE Trans. on Information Technology in Biomedicine 1, März 1997, S. 55–60.
- [Meer 99] D. Comanicu, P. Meer: *Distribution Free Decomposition of Multivariate Data*. In: Pattern Analysis and Applications, 1999.
- [Microsoft] *Trust-Based Security for Java - Microsoft Technologies for Java*.  
<http://www.microsoft.com/java/security/default.htm>
- [Multimedica] *multimedica - der Onlinedienst für Ärzte und Apotheker*.  
<http://www.multimedica.de/>
- [NASA] *NASA Telemedicine - History*.  
<http://www.hq.nasa.gov/office/olmsa/aeromed/telemed/history.html>
- [Najork 91] M. A. Najork, S. M. Kaplan: *The Cube Language*. In: 1991 IEEE Workshop on Visual Languages, Kobe, Japan, Oktober 1991, S. 218–224.
- [Numerical Python] *SIG for Built-in Matrix Types in Python*.  
<http://www.python.org/sigs/matrix-sig/>
- [OCOCAT 99] *OCOCAT-S: Objectcode and Optimizing Compiler Analyzing Tool*. Studie im Auftrag des Bundesamtes für Sicherheit in der Informationsverarbeitung, 1999.

- [OJI] *mozilla.org*:  
<http://www.mozilla.org/oji/>
- [OMG] *Object Management Group Homepage*.  
<http://www.omg.org>
- [OpenDX] *OpenDX Home Page*.  
<http://www.opendx.org>
- [Orfali 96] R. Orfali, D. Harkey, J. Edwards: *The Essential Client/Server Survival Guide. Second Edition*. John Wiley & Sons, 1996.
- [Ousterhout 98] J. K. Ousterhout: *Scripting: Higher Level Programming for the 21st Century*. In: IEEE Computer magazine, März 1998.
- [Pearlman 96] A. Said, W. A. Pearlman: *A new fast and efficient image codec on set partitioning in hierarchical trees*. In: IEEE Trans on Circuits Syst. Video Tech., Vol. 6, Juni 1996, S. 243–250.
- [Poremba 98] C. Poremba, N. Pickhardt: *Ökonomische Evaluation der Telepathologie*. In: Pathologe (19), 1998, S. 318–324.
- [Prechelt 00] L. Prechelt: *An empirical comparison of C, C++, Java, Perl, Python, Rexx, and Tcl for a search/string-processing program*. Technical Report 2000-5, 34 Seiten, Universität Karlsruhe, März 2000.
- [Price 93] B. A. Price, R. M. Baecker, I. S. Small: *A Principial Taxonomy of Software Visualization*, In: JVLC, Vol. 4, No. 3, Sep. 1993, S. 211–266.
- [Python] *Python Language Website*.  
<http://www.python.org>
- [RBerger 97] Roland Berger & Partner GmbH: *Telematik im Gesundheitswesen - Perspektiven der Telemedizin in Deutschland*. Bundesministerium für Bildung, Wissenschaft, Forschung und Technologie und Bundesministerium für Gesundheit. München, August 1997.

[RMI IIOP] *RMI-IIOP Programmer's Guide.*

[http://java.sun.com/products/jdk/1.3/docs/guide/rmi-iiop/rmi\\_iiop\\_pg.html](http://java.sun.com/products/jdk/1.3/docs/guide/rmi-iiop/rmi_iiop_pg.html)

[ROBODOC] *ISS.*

<http://www.iss.com>,  
<http://www.robodoc.com/>

[Redlich 96] J. P. Redlich: *Corba 2.0. Praktische Einführung für C++ und Java.* Addison-Wesley, München, 1996.

[Riede 99] U. N. Riede, H. E. Schaefer: *Allgemeine und spezielle Pathologie.* Thieme, Stuttgart, 1999.

[Rodgers 97] P. J. Rodgers, H. King: *A Graph Rewriting Visual Language for Database Programming.* The Journal of Visual Languages and Computing, 8(6), Dezember 1997, S. 647–674.

[Römer 97] W. Römer: *Entwicklung einer Arbeitsoberfläche zur Konformationssuche von organischen Molekülen.* Diplomarbeit, Universität Heidelberg, 1997.

[SWIG] *SWIG Home.*

<http://www.swig.org/>

[SWIG Tutorial] *SWIG Tutorial.*

<http://www.swig.org/Tutorial>

[Sauer 99] S. Sauer, G. Engels: *Extending UML for Modeling Multimedia Applications.* In: 1999 IEEE Symposium on Visual Languages, Tokyo, Japan, September 1999.

[Schiffer 98] S. Schiffer: *Visuelle Programmierung. Grundlagen und Einsatzmöglichkeiten.* Addison-Wesley, 1998.

[Schlag 99] G. Grasczew, G. Bellaire, S. Rakowsky, F. Engel-Murke, D. Steinnes, P.M. Schlag: *OP 2000.* In: Jäckel (Hrsg.): *Telemedizinführer Deutschland*, Ausgabe 2000, Bad Nauheim, 1999, S. 291–296.

- [Schneider 99] B. Schneier: *Applied Cryptography*. John Wiley & Sons, 1996.
- [Schroeder 97] W. Schroeder, K. Martin, B. Lorensen: *The Visualization Toolkit : An Object-Oriented Approach to 3-D Graphics*. Prentice Hall, 1997.
- [Schröder 99] J. Schröder: *Aufbau eines Systems zur Messung des Einflusses von niederfrequenten elektrischen Wechselfeldern auf die Proliferation und Motilität von Tumorzellen unter Einwirkung des Zytostatikums Epirubicin*. Diplomarbeit, Universität Heidelberg, 1999.
- [Rawolle 99] J. Rawolle, C. Lassahn, M. Schumann: *Wege zur Absicherung eines Intranets*. In: Informatik-Spektrum 22, 1999, S. 181–191.
- [Schurr 94] A. Schurr: *Rapid Programming with Graph Rewrite Rules*. Proceedings USENIX Symposium on Very High Level Languages (VHLL), Santa Fe, Oktober 1994, S. 83–100.
- [Shapiro 93] J. M. Shapiro: *Embedded image coding using zerotrees of wavelet coefficients*. In: IEEE Transactions on Signal Processing, Vol. 41, Dezember 1993, S. 3445–3462.
- [Smith 75] D. C. Smith: *Pygmalion: A Creative Programming Environment*. Ph.D. Thesis, Technical Report STAN-CS-75-499, Department of Computer Science, Stanford University, 1975.
- [Stotzka 96] R. Stotzka: *Automatisches Tumorklassifikationssystem für mikroskopische Gewebeschnitte - mit hybriden neuronalen und statistischen Klassifikatoren*. Shaker Verlag, 1996.
- [Strauss 99] J. S. Strauss, C. L. Felten, D. H. Okada, A. M. Marchevsky: *Virtual microscopy and public-key cryptography for Internet telepathology*. In: Journal of Telemedicine and Telecare 5(2), 1999, S. 105–110.
- [Sun] *java.sun.com - The Source for Java(TM) Technology*.  
<http://java.sun.com>

- [TIE] *Telemedicine Information Exchange (TIE) Home Page.*  
<http://tie.telemed.org/>
- [TelePath 97] G. Grimes, S. McClellan, J. Goldman, D. A. Conner, S. McClellan, G. Vaughn, T. Winokur, W. Fleming, J. McDonald: *Applications of Virtual Reality Technology in Pathology.* In: Proc. Medicine Meets Virtual Reality 5, San Diego, CA, USA, Februar 1997.
- [Telenor] *PathSight Information Brochure.* Second Opinion Solutions AS, P.O. Box 6701 St. Olavs Plass, N-0130 Oslo, Norway.
- [Tympel 95] V. Tympel: *Ein neues Bildaufnahmesystem für die konventionelle Lichtmikroskopie.* In: F. Hering (Hrsg.): *Mustererkennung 1996*, Springer Verlag, 1996, S. 543–550.
- [Ulsenheimer 00] K. Ulsenheimer, N. Heinemann: *Rechtliche Aspekte der Telemedizin – Grenzen der Telemedizin?* In: Jäckel (Hrsg.): *Telemedizinführer Deutschland*, Ausgabe 2000, Bad Nauheim, 1999, S. 130–137.
- [W3C] *W3C - The World Wide Web Consortium.*  
<http://www.w3c.org>
- [WebPath] *WebPath: The Internet Pathology Laboratory.*  
<http://www-medlib.med.utah.edu/WebPath/webpath.html>
- [Weinberg 96] D. S. Weinberg, F. A. Allaert, P. Dusserre, F. Drouot, B. Retallau, W. R. Welch, J. Longtine, G. Brodsky, R. Folkerth, M. Doolittle: *Telepathology diagnosis by means of digital still images; an international validation study.* In: *Human Pathology* 27(2), 1996, S. 111–118.
- [Weinstein 90] R. Weinstein, K. Bloom, and L. Rozek: *Static and dynamic imaging in pathology.* In: *IEEE Proc. Image Manage. Comm.*, Vol. 1, 1990, S. 75–85.
- [Wheeler 99] T. Wheeler: *Telepathology's slow start.* In: *Telemedicine Today* 7(1), Februar 1999, S. 37–39.

[Wootton 95] R. Wootton, D.R. Springall, J.M. Polak: *Image analysis in histology. Conventional and confocal microscopy*. Cambridge University Press, 1995.

[Zhao 94] B. Zhao: *Erfassung der strukturellen Heterogenität menschlicher Mammatumoren durch Erkennung und Klassifizierung spezifischer Wachstumsmuster*. Dissertation, Universität Heidelberg, 1994.

[VTK] *Kitware Home*.  
<http://www.kitware.com/>



## Danksagung

An erster Stelle möchte ich meinen Professoren Herrn Professor Komitowski und Herrn Professor Männer danken, die mir die Möglichkeit gaben, diese Doktorarbeit durchzuführen. Professor Männer ist es gelungen mit seiner Kompetenz und Unkompliziertheit einen Lehrstuhl aufzubauen, der an Kreativität und Atmosphäre seinesgleichen sucht.

Allen Mitarbeitern der Abteilung Histodiagnostik und pathomorphologische Dokumentation möchte ich für das angenehme Arbeitsklima danken.

Besonderes danken möchte ich Dr. Axel Bischoff und Dr. Fouad Kattit für die freundschaftliche Zusammenarbeit während unserer gemeinsamen Zeit am DKFZ. Ebenfalls danke ich Wolfgang Römer, Jörg Mattes und Jochen Schröder, die mit ihren Diplomarbeiten zum Entstehen dieser Arbeit beitrugen. Die Zusammenarbeit mit Jörg Mattes ist mir aufgrund seiner zahlreichen guten Ideen, der kompetenten Umsetzung und zahlreicher Diskussionen in sehr guter Erinnerung geblieben.

Auch Dr. Jürgen Hesser möchte ich für seine Unterstützung bei dieser Arbeit danken. Sein enormes Wissen und seine Fähigkeit, komplexe Zusammenhänge in kürzester Zeit zu verstehen und das Wesentliche herauszufiltern, haben mir bei so manchen Problemen geholfen. Markus Schill danke ich für die Unterstützung am Anfang des Projekts.

Ferner möchte ich meinen Korrektoren Dr. Jürgen Hesser, Dr. Axel Bischoff, Uli Kaps und Szilvia Deminger für die kritische Durchsicht dieser Arbeit und für die konstruktive Kritik danken. Meiner Frau Szilvia Deminger danke ich für die Geduld und für das Verständnis, das sie während meiner Promotion aufbringen mußte.

Bei meinem Zimmergenossen Alexander Gröpel danke ich für die angenehme gemeinsame Zeit, so manchem Latex-Tip und für zahlreiche interessante Diskussionen.

